

The Albelda Clayton-Matthews Paid Family and Medical leave Simulator Model Documentation

A report and simulation model presented to the Women's Bureau of the Department of
Labor

by

Alan Clayton-Matthews and Randy Albelda

November 10, 2015

Alan Clayton-Matthews
Associate Professor
School of Public Policy and Urban
Affairs
Northeastern University
Boston, MA 02115
a.clayton-matthews@neu.edu
617-373-2909

Randy Albelda
Professor
Economics Department and Public
Policy Ph.D. Program
University of Massachusetts Boston
Boston, MA 02125
randy.albelda@umb.edu
617-287-6963

This workforce product was funded by a grant (WB-26510-14-60-A-25) awarded by the U.S. Department of Labor's Employment and Training Administration. The product was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The Department of Labor makes no guarantees, warranties, or assurances of any kind, express or implied, with respect to such information, including any information on linked sites and including, but not limited to, accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability, or ownership. This product is copyrighted by the institution that created it. Internal use by an organization and/or personal use by an individual for non-commercial purposes is permissible. All other uses require the prior authorization of the copyright owner.

© Copyright 2015 by Randy Albelda and Alan Clayton-Matthews. All rights reserved.

I. Introduction

The basic strategy behind our approach to estimating the cost of a paid leave program in Massachusetts was to, as much as possible, base estimates of program costs on actual known leave-taking behavior, and where this was not possible, to estimate a range of program costs reflecting a range of reasonable assumptions about unknown aspects of behavior in the presence of a paid leave program. We wanted to be able to estimate the sensitivity of program costs estimates to these assumptions. We also wanted to be able to analyze the distribution of program benefits by demographic characteristics.

Furthermore, we wanted users to be able to estimate the costs of similarly structured paid leave benefit programs in other states, to be able to have some control over the assumptions about behavior that affect program cost estimates, and to be able to undertake their own distributional analyses.

We chose a simulation strategy as the best way to accomplish these goals. To obtain the best estimates possible about known leave-taking behavior, we use the Public Use Family and Medical Leave survey data collected by Abt Associates in 2012 for the Department Labor (referred to here as the DOL Survey) (McGarry, et al, 2013) to estimate behavioral models of leave-taking behavior conditional on the demographic characteristics of individuals, and use the Census Bureau's American Community Survey Public Use Microdata Sample (hereinafter referred to as the ACS or ACS PUMS) to predict leave-taking behavior conditional on the demographic characteristics of individuals.

The DOL Survey is the best available source of information on leave-taking behavior. It is a representative national sample of leave takers, leave needers (those persons who said they needed but did not take a leave), and other workers who did not take a leave. The survey, which was conducted between February and June 2012, includes extensive information on the number and types of leaves taken, how long they were, whether and to what extent the employer provided pay while on leave, and whether or not some or additional pay while on leave would result in a decision to take a leave or to have taken a longer leave. The survey includes several demographic characteristics

related to leave-taking behavior, including sex, race and ethnicity, marital status, the presence of children, education, family income, and whether or not the respondent was paid on an hourly basis. The survey is used to estimate several aspects of leave taking behavior, conditional on demographic characteristics and leave type. These include the probability of needing a leave, of taking a leave, of getting paid for a leave, of extending a leave if some or more pay were received, etc.

The ACS is a large national representative sample of persons. It is of sufficient size to obtain reliable estimates of paid leave program costs and of the distribution of program benefits at the state and sub-state level. The 5-year ACS PUMS can yield reliably accurate estimates at geographic areas consisting of one or more PUMAs (a PUMA is a geographic area that consists of a population of roughly 100,000 persons). This survey also provides a rich array of demographic characteristics that closely match those on the DOL Survey, which means that the behavioral models estimated on the DOL Survey can be used to predict leave-taking behavior on the ACS.

The simulation model is a software application that “runs” each sample person from the ACS through the estimated behavioral models and sets of assumptions about leave-taking behavior. The flow of the person through the software mimics the sequence of decisions and events that a person makes and experiences in the leave process. This is an appealing aspect of simulation methodology since its structural approach helps identify what assumptions are necessary in developing program cost estimates and at the same time clarifies the impact of these assumptions on the bottom line estimates.

At several points during the simulation, such as when a person decides to take a leave of a particular type or not, a decision is made based on a logit behavioral equation. The logit equation estimates the probability of deciding “yes”. This probability, which is a function of the person’s demographic characteristics, is compared to a random draw from a standard uniform distribution (any point on the number line between zero and one is equally likely to be chosen), and the random draw is compared to the probability given by the behavioral equation. If the random draw is less than this probability (or less than or equal, it really does not make any difference), the decision is “yes”, if not, “no”. The

model flow then directs the person to the next point in the modeling sequence, depending on the result of this random draw. This is the essence of simulation.

After each person has been passed through the entire flow, the result is a history of leave-taking behavior for a one-year period. The model generates micro data output files consisting of records for each sample person and leave taken. These files can be analyzed with standard statistical software or database applications.

Aside from errors related to the DOL survey and estimates of the behavioral equation parameters, there are two sources of statistical error related to the simulator that are important to consider. One is sampling error due to the ACS. The ACS is a sample and is subject to sampling error that affects program cost estimates. The magnitude of this error is approximately inversely proportional to the square root of sampling size, and can be reduced by concatenating successive years of the ACS together. The second source of statistical error is due to the simulation methodology itself when the dependent variable is binary (or categorical). Even if the coefficients of a behavioral equation are “correct”, individual predictions are not at the individual level. For example, suppose a logit equation predicts that the probability of taking a leave is 30 percent for a person with a certain set of demographic characteristics. For any *single* person, the simulation results in either the person taking the leave – a simulation error of 70 percent-- or the person not taking the leave – a simulation error of 30 percent. The law of large numbers assures that the error approaches zero on average as the number of persons “run” through this equation approaches infinity. The magnitude of this simulation error is inversely proportional to the square root of the number of “runs” through the equation. The incidence of some types of leave is small enough that this source of error is not negligible. This type of error can be reduced by concatenating ACS data files, but there is also another way to reduce simulation error. That way is to “clone” the sample ACS person (i.e., to create several duplicates of the same person) and to run each duplicate person through the simulation. The software allows the user to specify this option. At the state level, the ACS sample is large enough so that cloning is not really necessary; but for estimates at sub-state geographies, cloning may be an excellent way to reduce simulation error.

The next major section of this document describes the basic modeling strategy used by the simulation model. The third major section of this document describes how to use the simulation model. It contains three subsections. The first gives instructions on how to install the software on a computer, and describes the organization of the folders and what they contain. The second describes how to run the application and how to form the commands that control the model's parameter "handles" and direct the model's input and output files. These parameters allow the user to change the paid leave program's specifications, and also to change assumptions about participation and leave extension behavior. The third describes the output files created by the model and how to use statistics or database software applications to access these files. The fourth major section describes the flow of the model.

II. Modeling Strategy and Assumptions

The principal strategy behind the implementation of the model is to use econometric estimates of known leave-taking behavior when possible, and to incorporate reasonable assumptions and user-supplied options about unknown behavior. As new knowledge about behavior becomes available, the user may be able to incorporate that in model options – for example, new knowledge about take-up rates. In addition, new knowledge may be incorporated as it becomes available in future versions of the model.

Modeling Known Behavior

The best source of information on which to model several aspects of known behavior – the incidence of taking or wanting to take a leave of a particular type, the probability of receiving pay while on leave and the amount of pay received, the length of leaves taken, and the probability of meeting the eligibility requirements of a proposed paid leave program – is the Family and Medical Leave in 2012 survey by Abt Associates for the Department of Labor (McGarry, Klerman, Daley, and Pozniak, 2013). The population surveyed consisted of adults 18 and older who had worked for pay in the last 12 months. They were asked about leaves taken or wanted during the prior 18 months for reasons of own health or disability (including maternity disability); to care for a new child; for health conditions of children, spouses, parents, other relatives, and non-relatives; and for issues arising from the deployment of a military member. Due to small

sample sizes for some categories, we limited our analysis and modeling to the following six leave types:

1. Own health;
2. Maternity disability;
3. Care for a new child;
4. Ill child;
5. Ill spouse; and
6. Ill relative.

The sample of persons surveyed can be classified into four groups depending on whether they took a leave or wanted to take a leave or not:

1. Those who took a leave and who did not not take a leave they wanted to take (leave takers only, N=1,133);
2. Those who wanted to take a leave but did not take any leaves (leave needers only, N=219);
3. Those who did not take a leave or want to take a leave (employed only, N=1,301); and
4. Those who both took a leave and also did not take a leave that they wanted to take (dual takers/needers, N=199).

The sample was weighted to the population so population rates and totals could be inferred from the sample. The survey asked about the longest and most recent leaves taken or wanted – and the reason for that leave – in the last 18 months, whether those leaves were taken or wanted in the last 12 months, and how many leaves in all were taken in the last 18 months and in the last 12 months. Leaves were counted by “reason”, so intermittent leaves for a single reason were counted as a single leave. Leave takers were asked about the reasons and lengths of leave for up to two leaves: the longest and the most recent (often they are the same). Leave needers were asked about the most recent leave needed and the reasons for up to two more leaves needed. For both taker and needer

leaves, respondents were asked if they saw a doctor or had a hospital stay. For the most recent leave taken or needed, additional information was asked. For leave takers, this included questions about pay received while on leave; and if full pay was not received, whether they would have taken a longer leave if they had received additional pay. For leave needers, this included a question about why they didn't take the leave. Many respondents volunteered that they couldn't afford to take an unpaid leave. These questions about additional pay and affordability were helpful in modeling the response of leave lengths and participation in the presence of a paid leave program. Leave takers were also asked about whether some of the pay received while on leave was part of a TDI program or a state family leave program.

Respondents were also asked about their work. Particularly useful for modeling behavior and estimating program eligibility were questions about weekly hours, whether they worked full year and were continuously employed by a single employer, how many employees worked at their organization within 75 miles, and whether they were paid on an hourly basis or not. Demographic information on respondents included age, sex, race/ethnicity, marital status, educational attainment, family income, and how many children were in their care.

Earlier theoretical work and statistical analysis of a prior Department of Labor family leave survey (Westat, 2001; Albelda, and Clayton-Matthews, 2010) established that the information in this survey would be useful in estimating statistical models of the probability of taking or needing a leave of a particular type, the probability of receiving partial or full pay, and the probability of meeting the eligibility requirements of the FMLA law or a proposed paid leave program. The estimation strategy involved a specification search that began with a full set of demographic and economic variables and "tested down" to a specification that included independent variables that were at or near statistical significance at the 5% level and that "made sense" in terms of yielding estimated coefficients of the expected sign and reasonable magnitude. The estimated relationships appear in the model source code (in the Parameters class) and may be reported in a separate publication.

These statistical models are implemented in the model by applying the estimated coefficients to variables on the ACS for each sample individual worker. Most of these models estimate a probability: the probability of taking or needing a leave for a particular reason, the probability of receiving pay while on leave, the conditional probability that that pay was full pay, etc. Using the coefficients of the logit regression model and applying them to the sample individual's independent variables yields a probability of taking or needing a leave, of receiving pay while on leave, of receiving full pay conditional on receiving any pay, etc. These probabilities are compared to a random draw from a standard uniform distribution – using the model's pseudo random number generator – to determine whether an outcome happens or not. Other models – for example, for fraction category of pay received, or number of leaves taken – are estimated by an ordered logit model, and the random draw determines the category by the estimated cumulative probability distribution of outcomes. Several models – usually when sample sizes are too small to estimate probabilities conditional on observable characteristics – are simply the weighted distributions from the survey. These models are identical to statistical models that contain only a constant, and are handled by the simulator in the same manner as other models that predict probabilities of binary outcomes or ordered outcomes.

This strategy “works” because the both the DOL survey and the ACS are representative samples (after weighting) of the population and both contain closely similar measures of independent variables. The match of variables is not complete, however, so a few variables not available on the ACS have to be imputed. For behavioral models these involve two variables: whether or not the worker is paid on an hourly basis, and whether or not the worker is covered and eligible under the FMLA law. This eligibility criterion is significant in several behavioral relationships, and involves weeks worked, worked full time continuously for a single employer in the past 12 months, and worked for a firm that had at least 50 employees within 75 miles. Other eligibility requirements of proposed paid leave programs might require knowing weeks worked; and benefit rules of proposed programs usually pay benefits proportional to weekly earnings.

The ACS does not ask whether pay is received on an hourly basis; does not ask about employer size; does not ask about the number of employers that the person worked

for in the last 12 months; does not ask about weekly pay; and records weeks worked in aggregated categories. These variables are imputed on the ACS using models and distributions estimated from the Current Population Survey (CPS) – which *does* include these variables – conditional on demographic and economic variables common to both the CPS and ACS surveys. These models and their estimated parameters are included in the Wage class in the software. Weekly wages on the ACS are estimated as annual earnings divided by the imputed number of weeks worked.

Simulating Unknown Behavior

Some information about leave-taking behavior needed for our simulation procedure cannot be estimated from the DOL 2012 survey, although some information collected there is useful in making some reasonable assumptions. The three main pieces of unknown information – whether a worker will use a paid program or employer benefits; program take-up rates; and whether a worker will extend a leave in the presence of a program – are discussed below.

1. How employer benefits affect participation in paid program.

The decision to participate in the paid leave program, given that a person is eligible, will in large part be based on the level of program benefits the worker would receive compared to the next best alternative. These alternatives consist of employer pay (if the person receives it) or nothing (if the leave is unpaid in the absence of the program). In order to compensate for the time and effort of applying to the program, program benefits would have to exceed the next best alternative by some amount. This amount may differ systematically by income and by other factors. It may also vary randomly across different individuals, and even for the same individual, at different times.

In the model, this participation decision is implemented by an arbitrary logit equation with two independent variables: the difference between weekly paid program benefits and weekly pay received while on leave, and family income. The participation probabilities it yields are given in Table 4 for several combinations of benefit/pay differentials and family income.

2. *Take-up Rates*

The simulation model estimates the number of all eligible workers that would use a paid leave program in light of current employer benefits. This estimate assumes that everyone taking a leave knows about the program and that the program is virtually costless to use. That is, the output from the simulator assumes a 100 percent take-up rate. However, this is completely unrealistic, which is why one of the policy parameters that can be adjusted by the user is take-up rates. The degree to which eligible leavers might use a paid leave program depends on a variety of factors beyond the scope of what can be uniformly modelled or assumed. Three important ones are: general knowledge of the program by workers; administrative complexity in obtaining program benefits; and workplace culture that either encourages or inhibits use. Recent experiences with care and bonding leaves in California, New Jersey and Rhode Island suggest that take-up rates, at least for several years, will be low. A recent estimate indicated that 25-40 percent of new mothers used the 6 week care leave in California, even after 10 years of implementation. Appelbaum and Milkman found that fewer than 50 percent of California workers knew about paid family leave. The degree to which state administrators and paid FML advocates work to make the program known will positively affect take-up rates. Use of any program will require time on the part of leave takers (and employers) to fulfill the administrative requirements of the leave. An easy-to-use program can reduce that time (which is a cost). Still, workers that take relatively short leaves may not bother at all. There may be other real or perceived costs to taking a program leave. If workers fear their position at their job might be threatened if they take a leave, then take-up rates will be low. For example low-wage workers may fear being replaced altogether while high-wage employees may fear an employer might not provide them with better opportunities.

The simulation model lets the user select a take-up rate based on some reasonable assumptions about the percent of eligible workers that might use a particular program in a particular state. The user can apply different take-up rates for different kinds of leaves. For example, there may be reason to believe that maternity disability leaves might have higher take-up rates than other leaves. Almost all mothers that give birth do leave work for a continuous period of time that is usually known in advance. Employers and employees typically expect new mothers to be away from work for more than a few

weeks. Further, obstetricians and others in pregnant mother's networks are likely to inform them of a paid leave program so usage might be higher than other types of leaves.

3. Extending a leave in the presence of a program

In the presence of a paid leave program, leaves would not be shorter than in the absence of the program, but they may be longer. Lacking empirical evidence about the effect of program benefits on extending leave lengths, we estimate the probability of extending a leave. Because this decision is complex and affected by length of leave before the decision to extend, availability of employer pay, and whether the leave is job-protected, we use different extension rules in the simulation. For workers with short leaves (leaves that end before the waiting period of the program is over), we estimate the probability of taking a longer leave using logit regression estimations relying on the response to the DOL survey question, "Would you take a longer leave if you received some/additional pay?" If the model simulates an extension, we arbitrarily extend the leave for 1 week. We assign a different decision to those employees who reach the end of their original leave length (the length they would take if there was not program) and are receiving either program or employer benefits (but not both). We assume the probability of extending these leaves using program benefits are 25 percent and for those who do extend, that the extension is equal to 25 percent of their original length, not to exceed the maximum length of the program. The last decision applies only to those who have exhausted the paid program and still have some employer benefits available to them (based on the simulation). In this case the simulator assigns them a 50 percent probability of taking an extended leave for as long as they still have employer benefits. In all cases, if the original length of leave is less than the FMLA job-protection length of 12 weeks, an option in the model allows the user to restrict the leave to a maximum of 12 weeks. In the case of own health leaves, there was a significantly longer distribution of leaves for workers who received some part of their pay from state programs; and so the model incorporates longer own health under a paid leave program by using the distribution of leaves experienced by these covered workers.

III. How to Use the Model

How to Install the Model

The model comes in a two-level folder hierarchy. The main folder, named “FML2” (called the FML folder) contains the simulation executable programs and subfolders that contain the documentation, software, parameters, input and output files. These are named “Documentation”, “Software”, “Parameters”, “Input”, and “Output” respectively.

Copy the FML folder and its contents to your computer’s internal hard drive. It can be placed anywhere in a user’s directory structure that is convenient. It is important to maintain this relative directory structure, since the relative paths for the input, output, and parameter files are “hard coded” into the software.

System Requirements

Windows 7 or later operating system with Service Pack 1, Microsoft Visual Studio Express 2015 for Windows Desktop installed (a free download from Microsoft is at: <https://www.visualstudio.com/products/mt238358>), 8GB RAM, 30GB of hard drive space. The Visual Studio Express 2015 is needed because it installs certain required operating system files. The hard drive space needed depends on how many ACS input files you need. Thirty GB is enough to hold the 5-year U.S. ACS PUMS 2009-2013 data set as well as a few state ACS data sets.

The Directory Structure

The executable programs are in the FML folder. This version of the model contains two executables: “FML2_State_2009-2013.exe”; and “FML2_US_2009-2013.exe”. The former runs on a 5-year 2009-2013 ACS PUMS dataset, and the latter runs on the 5-year U.S. 2009-2013 ACS PUMS dataset. The subfolders and their contents are described in outline form below.

Documentation: This folder contains this documentation, the command reference appendix as a separate document, and the PUMS data dictionary from the U.S. Census Bureau.

Input: This folder contains the ACS files that serve as input files for the simulator, as well as the command and other user-created input files described in the command reference appendix. This folder may already contain the 5-year U.S. ACS PUMS 2009-2013 files (8 files in all in .csv format) and/or a state 5-year ACS dataset. Download any 5-year ACS PUMS 2009-2013 files you would like to use directly from the Census Bureau to this folder. The model uses the “.csv” formatted files from the Census Bureau, which can be obtained at <http://www.census.gov/programs-surveys/acs/data/pums.html> . This link is also available from the American Community Survey’s home page <http://www.census.gov/programs-surveys/acs/> . From this home page, choose the “Data” link, and then the “PUMS Data” link. There may be an example command file (cmd.txt) included in this input folder, as well as other user-supplied sample files that are referenced in the command file.

Output: This folder will contain the output files produced by the simulation. If you plan to run several simulations and you want to keep a record of what you did, it is recommended that you manage your output files, perhaps by copying them after each run into a subfolder of this folder, or to some other place.

Parameters: This folder contains several text files of parameters that are read by the simulator when it starts up. Do not change these files! It is recommended that you do not touch this folder. If you would like to read or open the files in this folder, it is recommended that you change the files’ properties to “ready only” so you do not accidentally change them.

Software: This folder contains the source code for the simulator. These files can be opened and inspected in any text editor. They are not used by the simulator, so you can alter them – for example, by putting in comments – as you wish. This software contains the code for the simulator’s procedures, so that an experienced C++ programmer can “look inside” the simulator’s engine. However, it does not contain the code for the objects and containers used by the model, for example, the code that implements containers like matrices and arrays, objects that handle dates, code that parses commands, etc.

How to Run the Model

To start the simulator, double click the “FML2_State_2009-2013.exe” or the “FML2_US_2009-2013.exe” file icon in the FML folder. Your system may not display the “.exe” extension. The simulator will open a window and prompt you for the command file name. Include its extension (“.txt”) as well.

The command file tells the simulator what input and output files it will use, what level of detail should be output, what ACS variables to include in the output, whether cloning should be used, what the program eligibility requirements and program benefit rules are, and what options the simulator should use that affect program participation and leave lengths. All input files should reside in the Input folder before the simulator is run. All output files will be written to the Output folder, over-writing any files of the same name. The commands, what they do, their syntax, and examples are given in the Command Syntax appendix. A sample command file is given in Appendix A.

How to Analyze the Model Results

The output files that contain results of the simulation are all in comma-separated format, with variable names in the first row, and so can easily be read by statistical and database software packages. Two files are also created to aid in processing the output. One is the documentation file, which contains information on all the variables in the output files. For Stata users, another file, called “labels.do”, is also created in the Output folder. It contains Stata label commands for each variable.

Identifier variables are output to each file to enable merging of information from one file to another. The “personid” can be used for merging the “main” and “leaves” files. The “personid” and “leaveid” variables are available on every file except “main” (since “main” is at a more aggregate level than a leave). The “nstate” variable can be used, along with “personid” and “leaveid”, to link the three state files together (the “states”, “weekly benefits”, and “weekly employer payments” files). Finally, “weekid” in the “weekly employer pay and program benefits” file, “benwkno” in the “weekly benefits” file, and “paywkno” in the “weekly employer payments” file all refer to the same week within each leave, allowing these files to be merged if desired. Appendix B

provides a complete list of variables names, label, level, and data source for each output file.

Each file also contains a variable called “weight” that should be used to weight records up to population totals. The variables from the ACS maintain the same spelling as in the official ACS data definition files, so those files, included in the Input folder, can be used for variable codes. Several date variables come in two forms: 1) a triplet of variables recording the month, day, and year; and 2) the number of days since January 1, 1960. This latter form is used by Stata, which makes handling of dates convenient in Stata.

The key variable for calculating aggregate paid leave program benefit costs is the variable “benamt” on the “leaves” file. The weighted sum of this variable gives total program benefit costs.

Technical Support

For technical support, contact Alan Clayton-Matthews at a.clayton-matthews@neu.edu , 617-373-2909.

IV. What the Model Does: The Flow of the Model

This section first provides programming vocabulary and then describes what the simulation does by following the flow through the model’s software. For the most part, this flow corresponds to the timing of decisions and modeling of behavior individuals make and exhibit in the process of taking a leave for personal or family-related medical reasons. Again, the way in which this simulator models the leave process, including the simulated behavior and personal decisions, are highly influenced by, and constrained by information and the structure of the DOL survey.

Some Programming Vocabulary: Classes, Objects, Instances, and Procedures

The simulator is written in the programming language C++ (C Plus Plus), an object orientated language that facilitates the construction of complex software applications by allowing the programmer to break the problem down into a small number

of “objects” called “classes.” Each class is more or less self-contained and corresponds to a logical piece of the problem that usually represents a physical object (like a person or accounting ledger book), or a concept (like a set of benefit program rules or behavioral equations). A class is a template, consisting of source code, that describes the characteristics of an object and how it behaves, and an object is a particular “instance” of the class. The software code that describes the rules of how objects work is arranged into units of code called “procedures”. These are sometimes also called “methods”; and a Fortran programmer might call them “subprograms”. Each object of the same class has the same procedures, yet each object can – and almost always has – a different set of data. For example, each person who is run through the simulator is an object with different characteristics than other persons. Each person might have none, one, or more leave objects that describe the characteristics of each leave. Each leave might involve a different sequence of events, where each event is a “state” object.

A note on syntax:

We refer often to the source code document and procedure or class within the source code document. The syntax is as follows: the source code document file name appears in italics and the procedure appears as regular type, with the two separated by a comma. When the procedure belongs to a class, the class name precedes it, and is separated from the procedure name by a double colon. The whole reference is enclosed in parentheses. For example, (*benefitcalc.cpp*, BenefitCalc::MakeNamesBenefits) refers to the procedure “MakeNamesBenefits” in the class “BenefitCalc” in the source code file “benefitcalc.cpp”. Note that all source code files end in either the extension “cpp” or “h”. Input and output file names will also be italicized.

Initialization of Program Parameters and Behavioral Relationships

After the application prompts for file names and user-controlled program parameters (which are handled in (*fml2.cpp*, main)) the application creates an instance of the class named “Parameters”. There are three distinct sources the simulator uses to generate the information contained in “Parameters” class. The first source is user-

supplied program parameters (provided by the user and include the specifications of the paid leave programs).

The second source of information in the “Parameter” object comes automatically from files generated from tabulations of the *DOL Survey*. These file-read parameters are the distributions of leave lengths. The names of these files are listed in Table 1 and each can be found in the Parameters folder of the FML software application. Each type of leave has a different distribution of lengths, and except for own health, are different for men and women. Each file gives the cumulative distribution of leave length in days. The first row of each table gives the dimensions of the distribution matrix. The following rows contain the cumulative distribution. The first column gives the number of days, and the second column gives the proportion of leaves whose length is equal to or less than the corresponding number of days. These distributions are tabulated from the *DOL Survey*. Table 2 contains the information in the file “*FML/Parameters/length NEW CHILD women 2.txt*.”

Analysis of leaves lengths using the DOL Survey indicates that leave lengths of illness types are related to the severity of illness. However, aside from the gender of the leave taker (for all but own-health) and severity of illness, there are no other significant predictors of leave length. Importantly, whether or not the leave taker receives pay from his/her employer does not seem to be associated with the length of the leave. Since the ACS does not have information on individuals’ illnesses, the application simulates leave length by randomly drawing from the distribution that corresponds to the type of leave and gender of the leave taker.

The third source of information is initialized by the simulator when it creates the parameter object. This information concerns the amount of pay received while on leave. These estimates are from the earlier 2000 DOL Survey (Westat, 2001), since the newer survey did not ask these questions. Those leavers who indicated that they received partial pay from their employer while on leave were asked if they received at least some pay for each pay period that they were on leave (HA10D), and if not, was the pay for their full salary or only for a part of their salary (HA10E). Leavers were also asked what proportion of usual pay they received in total over the entire length of the leave (HA10F).

The responses to these questions were tabulated separately for each leave type, and expressed as conditional probabilities. They appear in (*params.cpp*, Parameters::Parameters). As an aid to reading these parameter values, the numbers for own health leaves are given in Table 3.

The Parameters class also contains the behavioral equations that estimate the probability of various events occurring, such as taking a leave, receiving pay, participating in the program, extending one's leave in the presence of the program, etc. These are described below as they are encountered in the flow.

The Main Program Loop

The application reads the ACS input file household by household, and within each household, passes each person through the simulator (*fml2.cpp*, main).

First, it is determined whether or not the person is an adult civilian who worked last year, and was not self-employed (*fml2.cpp*, main; *fml2.cpp*, filterRequirements). Only these persons are in the universe of possible leave takers and passed through the rest of the simulator.

Some necessary information is not directly available on the ACS, and therefore is estimated or simulated. These include weeks worked (imputed from the categorical weeks worked variable), weekly wage (annual earnings divided by weeks worked), paid hourly or not, employer size, and worked for a single employer last year or not. The models for these imputations are in the Wage class (*wage.cpp*, Wage::Wage), except for employer size, which is imputed in (*params.cpp*, Parameters::EmployerSize).

Based on these imputations, the simulator next determines the work and employer-size eligibility requirements for FMLA and for the paid leave program, using information on the person's work history. To approximate the work requirement under FMLA, the person had to have worked at least 1250 hours last year, and only have had one major employer last year (*params.cpp*, Parameters::EligibleWorkerFMLA). In addition, for eligibility coverage under FMLA, the size of the establishment must be at least 50 employees (*params.cpp*, Parameters::EligibleEmployerWorkerFMLA). This concept of FMLA eligibility under the work and employer size requirements is used as an

independent variable in several of the behavioral equations in the model, because it influences the person's ability and willingness to take a leave, and also is correlated with other personal and job characteristics that are not measured by other independent variables. Worker eligibility and employer coverage under the proposed program is calculated according to user-supplied eligibility requirements (*params.cpp*, `Parameters::Eligible`).

The person then enters the main software program loop illustrated in Figure 1. Each person is run through two branches illustrated in the figure (*fml2.cpp*, `set_leaves`). The person might be a leave taker, a leave needer, or both in a given year. On the left branch, the probability of a person's most recent leave being each of the six possible leave types is estimated conditional on the person's characteristics (*params.cpp*, `Parameters::PrTake`). These probabilities are compared to a draw from a standard uniform probability distribution. (Think of a "Wheel of Fortune", where the size of each slice on the wheel is proportional to the probability of a particular leave type, with the remaining large slice representing no leave.) Note: except where noted in Figure 1, each arrow represents a positive outcome. A negative outcome results in the person "dropping out" from taking a leave.

If one of the leave types is chosen, the possibility of more than one leave is simulated (*params.cpp*, `Parameters::PrMultipleLeaves`); and if so, the number of leaves greater than one is simulated as a random draw from the probability distribution of 2 through 6 possible leaves (*params.cpp*, `Parameters::PrDistMultipleLeaves`). The types of these additional leaves, if any, is simulated from an estimate of the conditional probability distribution of a second leave (conditional on a first leave) (*fml2.cpp*, `set_leaves`). This conditional probability distribution was estimated from those sample persons in the DOL survey who reported on the type of leave for both their longest and most recent leave, when these were different leaves. The survey implied that the probability of taking a second ill child leave or a second ill parent leave was higher than the unconditional probability of each, and the probability of taking a second maternity disability or new child leave in a given year was effectively zero.

The leave length is simulated as a random draw from the estimated distribution of each type of leave length given by the DOL survey. Except for own health leaves, these differed by sex, with women tending to take longer leaves than men. For own health leaves, leave lengths were longer for those who stated that they received some pay from a TDI or state paid leave program, so two leave length distributions were used: in the absence of a program, the distribution of leave lengths for persons who did not report receiving these payments was used; in the presence of program, the distribution of leave lengths for persons who did report receiving these payments was used.

The text files that contain these estimated distributions are noted in Table 1. Leave lengths are counted in days, ignoring weekends, so a leave of two weeks, for example, is ten days. At this point in the program flow, the leave lengths represent those in the absence of a paid leave program, except for those persons who would not have taken a leave in the absence of such a program. Later in the flow, in the presence of the paid leave program, the person may choose to extend their leave.

Up to this point, the simulation on the right branch, for leave needers, is similar (*fml2.cpp*, *set_leaves*), except that simulated leave lengths represent leave lengths *if* they were to take a leave. The models used on this branch include (*params.cpp*, *Parameters::PrNeed*), (*params.cpp*, *Parameters::PrMultipleNeeds*), and (*params.cpp*, *Parameters::PrDistMultipleNeeds*).

For leave takers, their weekly payments while on leave in the absence of a program is simulated in stages. First, whether or not they receive any pay while on leave (*fml2.cpp*, *processPerson*) and (*params.cpp*, *Parameters::PrPaidLeave*). Next, conditional on receiving pay, was it full pay (*fml2.cpp*, *benefits*), (*fml2.cpp*, *payGroup*), (*params.cpp*, *Parameters::PrFullPay*); and if not, what fraction of pay was received (*params.cpp*, *Parameters::OProbPayGroup*).

For those who were partially paid, the 2000 DOL Survey asked if the respondent received some pay for each pay period that they were on leave; and if not, in the pay periods for which they *did* receive pay, was it for their full salary? As described in the section on parameter initialization and illustrated in Table 3, the survey was used to estimate these conditional probability distributions for each leave type and payment

group (less than half pay, about half pay, more than half pay). If a person's leave was partially paid, their payment schedule was randomly selected from the corresponding conditional probability distribution for their leave type (*fml2.cpp*, *benefits*; *params.cpp*, *Parameters::PrSomePayCells*).

At this point, the application has determined if a person received some pay each week; and if not, if that person received full pay for some weeks; and if, over the course of their leave, a person received less than half of full pay, about half of full pay, or more than half of full pay. The weekly pay schedule is then filled out using arbitrary rules subject to these payment schedule and amounts constraints. For example, those persons who received some pay for each week of their leave, but who received less than one quarter pay in total, were assigned 12.5 percent of their weekly pay in each week of their leave (*fml2.cpp*, *benefits*), while those persons who received some pay each week, and more than three-quarters but less than full pay, were assigned 87.5 percent of their weekly pay in each week of their leave (*fml2.cpp*, *benefits*).

For leave needers, the model simulates whether they would take a leave if there were a paid leave program based on their reason for not taking a leave being that it was not affordable (*params.cpp*, *Parameters::PrTakeGivenProg*). If not, they are classified as an ultimate leave needer. If they do, they then follow the same remaining path as leave takers.

At this point, the leave taker's (or potential leave taker, if originally a needer), eligibility is determined (*fml2.cpp*, *set_leaves*). The work and employer eligibility conditions have already been determined by this point, so here it is determined whether or not they saw a doctor or went to a hospital (or whether the person they took a leave to care for saw a doctor or went to the hospital). These are computed by comparing the probability of a logit behavioral equation for each condition, i.e., seeing a doctor and going to the hospital (*params.cpp*, *Parameters::PrDoctor* or *params.cpp*, *Parameters::PrDoctorNeed*; *params.cpp*, *Parameters::PrHospital* or *params.cpp*, *Parameters::PrHospitalNeed*) to a corresponding random number. The doctor and hospital requirements vary somewhat depending on the leave type. Essentially, to be eligible for an FMLA-defined leave (except for new child) requires either seeing a doctor

or going to the hospital (*params.cpp*, `Parameters::EligibleDoctorHospital`), and it is presumed that if the person or the person they were caring for went to the hospital, they also saw a doctor.

After it has been determined what leaves, if any, the person takes, and their lengths, the leaves are then distributed across a calendar where their leave either finishes in a 12-month period beginning April 16, 2011 and ending April 15, 2012, or they are still on leave on April 15, 2012. The dates for the beginning and ending of the 12-month period are not critical. This period was chosen simply because the survey was conducted between February and June of 2012, so April 15 was approximately in the middle of this period. The possibility of having an unfinished leave at the end of the calendar is simulated by (*params.cpp*, `Parameters::PrUnfinished`). The random assignment to the calendar is given by (*Calendar.cpp*, `Calendar::AssignLeaves`). This assumes that leave-taking is not seasonal. Although the model simulates leave ending dates that are uniformly distributed throughout the year, it does not guarantee that for any simulated person, the dates “make sense” in that it is possible that two simulated leaves overlap in time. However, what it *does* achieve is a reasonable estimate for the extent to which some leaves which take place during a given year “spill” outside the yearly time period, either because they began before the year began or ended after the year ended.

Employer pay, program benefits, and leave length in the presence of a paid leave program

The next step in the model is to simulate employer pay, program benefits, and possible extensions of leave length in the presence of a paid leave program. The application simulates the sequence of events and choices that a leaver would reasonably experience, given their weekly leave history and weekly schedule of employer payments simulated up to this point, in the absence of a paid leave program. Three important, and reasonable, assumptions are embodied in this part of the simulation:

1. Not all eligible recipients participate in the program, due to lack of information, the hassle of applying, or other reasons. The proportion of eligible paid program leave takers who *do* participate is called the “take-up rate”, and is one of the program parameters set by the user. The user can specify a different take-up rate for each leave type.
2. The decision to participate in the paid leave program, given that the person is eligible, is also based on the level of program benefits he/she would receive compared to the next best alternative, which is employer pay if the person received it, or nothing, if the leave was unpaid in the absence of the program. In order to compensate for the time and effort of applying to the program, program benefits would have to exceed the next best alternative by some amount, and furthermore, this amount may differ systematically by income and by other factors. It may also vary randomly across different individuals, and even for the same individual, at different times. This participation decision is implemented by an arbitrary logit equation that has the difference between weekly paid program benefits and weekly pay received while on leave, and family income as independent variables. The participation probabilities it yields are given in Table 4 for several combinations of benefit/pay differentials and family income. The user can turn off this participation module if they wish, and rely solely on the take-up rate parameter instead (see section How to Run Model).
3. The part of the simulation flow that establishes employer pay, program benefits, and possible extensions of leave length in the presence of a paid leave program is illustrated in Figures 3 through 5. The software models the process as a sequence of states (i.e. section of source code with at least one path of entry or exit), represented as circles in the diagrams. Some of these states are “decision” states, in which the leaver must make a decision to participate in the program, or to extend their leave beyond its “original” length, the length simulated prior to this point. The transition from state to state, represented by arrows, is the result of events or decisions, such as the end of receipt of employer pay, the original length of leave being reached, the

decision to participate or extend a leave, etc. Diamonds represent predetermined conditions or conditions over which the person has no control, such as whether the person is eligible or receives employer pay. The flow from state to state traces a “path”. Table 5 lists each of the states as used in the source code referred to in Figures 3-5.

In order to illustrate how this part of the simulator works, let’s follow one of the many possible paths that a leaver might experience. Suppose that the person originally took a partially paid leave of 3 weeks from her employer, got paid 30 percent of her full time pay for the first two weeks of leave, and is eligible for the paid leave program. This person might take the following path.

The person begins in state number 0, “Begin employer leave, eligible for paid leave program” (*fml1.cpp*, *benefits*; *benefitcalc.cpp*, *BenefitCalc::BenefitCalc*; *benefitcalc.cpp*, *BeginPaidLeaveElig::BeginPaidLeaveElig*).

After the waiting period of a week, the person decides whether or not to participate in the program, and is in state number 8, “Participate in paid leave program?” (*benefitcalc.cpp*, *Participate_1::Participate_1*). The weekly benefit from the paid leave program substantially exceeds the person’s partial pay from her employer, so that the probability of participation in the program is high (*params.cpp*, *Parameters::PrParticipate1*). In fact, suppose that the probability is greater than the random number drawn, so that the person participates.

The person is then in state number 2, “Begin paid leave program” (*benefitcalc.cpp*, *BeginProgram::BeginProgram*).

After being on the program for two weeks, the original leave length is reached, and the person then decides whether or not to extend her leave. The person is in state number 12, “Extend?” (*benefitcalc.cpp*, *Xtend_1::Xtend_1*). The simulation sets the probability of extending the leave to 25 percent. (*params.cpp*, *Parameters::PrExtend1*). Suppose the random number drawn is less than .25, so the person extends her leave. The

leave is then extended by 3 days, or 25 percent of the original leave length of 15 days (*params.cpp*, *Parameters::DaysExtend*).

The person is then in state number 7, “Begin/continue paid leave program” (*benefitcalc.cpp*, *ContinueProgram::ContinueProgram*). The person remains in this state for three days, and then ends her leave, entering state number 20, “End leave” (*benefitcalc.cpp*, *EndLeave::EndLeave*).

The “End leave” state in the simulator performs some accounting and cleanup tasks (*benefitcalc.cpp*, *BenefitCalc::cleanup*). Among other tasks, a number is assigned to a variable called “path” that uniquely identifies this path that the person traversed. The “path” variable is calculated as the sum of a number of terms, where each term is the number two raised to the power given by the state number. The “End leave” state is omitted from this calculation since every leave ends in the “End leave” state. The value of “path” calculated for this particular path is 4485.

The processing for each person ends with outputting nearly all the information from the simulation to several files. Information from each leave is output to the “leaves” file, which contains the path traversed for each leave, along with summary information on program benefits received, employer pay received, the type of leave, its length in days, and other information. A record of the traverse through each state is also output to the “state” file, and the weekly record of program benefits and employer payments are written to three “weekly” files. The “main” file outputs information at the person level, including a summary of the leaves the person took, if any, and demographic information from the CPS so that the distribution of program benefits can be analyzed.

V. Conclusion

The application provides a sophisticated estimation of family and medical leaves. The program is relatively user friendly and includes considerable flexibility for the user in terms of possible paid leave program specifications and behavioral responses to a paid program. The application is a powerful estimation tool that can be used at the national, state, or even the metro area or city level. It not only estimates the amount that a paid leave program might cost, it also estimates the amount of employer benefits paid (in the absence and presence of a paid leave program) as well as employee uncompensated

wages due to any portion of earnings that are unpaid during a leave. Further, because the simulator outputs add the simulated variables onto the ACS, it is possible to look at the beneficiaries of paid leave programs by gender, race and ethnicity, marital status, and income level.

References

Albelda, Randy, and Alan Clayton-Matthews, 2010. *The Institute for Women's Policy Research and Labor Resource Center Paid Family and Medical Leave Simulation Model*. IWPR #A143. Institute for Women's Policy Research, Washington, D.C.

Klerman, Jacob Alex, Kelly Daley, and Alyssa Pozniak, 2014. *Family and Medical Leave in 2012: Technical Report* (September 7, 2012; revised April 18, 2014). Abt Associates, Inc., Cambridge, MA.

McGarry, Nancy, Jacob Alex Klerman, Kelly Daley, and Alyssa Pozniak, 2013. *Family and Medical Leave in 2012: Revised Public Use File Documentation* (August 23, 2013). Abt Associates, Inc., Cambridge, MA.

Westat, 2001. Family and Medical Leave 2000 Survey of Employees Codebook.

Table 1
Length of Leave Cumulative Distribution Files

length OWN HEALTH 2 no prog.txt
length OWN HEALTH 2 prog.txt
length MATERNITY-DISABILITY 3.txt
length NEW CHILD men 2.txt
length NEW CHILD women 2.txt
length ILL CHILD men 2.txt
length ILL CHILD women 2.txt
length ILL SPOUSE men 2.txt
length ILL SPOUSE women 2.txt
length ILL PARENT men 2.txt
length ILL PARENT women 2.txt

Table 2
Contents of “length of NEW CHILD women 2.txt”

10	2
1	0.054054
10	0.135135
25	0.162162
30	0.324324
40	0.378378
45	0.513514
60	0.648649
68	0.864865
90	0.972973
135	1

Note: the first row give the row and column dimensions. The rest of the rows contain the number of days on leave in the first column the cumulative probability distribution in the second column.

Table 3

For those who receive partial pay, how much and when was it received for those with own-health leaves

About how much of your usual pay did you receive in total?		Did you receive some pay for each pay period that you were on leave?	If not, when you did receive pay, was it for your full salary?
	Less than half	0.6329781	0.3273122
	About half	0.8209731	0.3963387
	More than half	0.9358463	0.3633615

Notes: Cells are proportions of respondents.
From variables HA10D, HA10E, HA10F of the DOL Survey

Table 4

Probability of Participating for Selected Values of Benefit/Wage Differential and Family Income

		Difference Between Weekly Program Benefit Amount and Next Best Alternative		
		\$ 25	\$ 50	\$ 125
Family Income	\$ 10,000	0.12	0.59	1.00
	\$ 20,000	0.08	0.48	1.00
	\$ 30,000	0.05	0.38	1.00
	\$ 40,000	0.04	0.28	1.00
	\$ 50,000	0.02	0.21	1.00
	\$ 60,000	0.02	0.15	1.00
	\$ 70,000	0.01	0.10	0.99
	\$ 80,000	0.01	0.07	0.99
	\$ 90,000	0.00	0.05	0.98
	\$100,000	0.00	0.03	0.98

These probabilities are based on a logit equation where the independent variables are the difference between the weekly program benefit and the next best alternative, and family income. The "next best alternative" is either the weekly pay received from the employer while on leave, or zero if the leaver receives no pay while on leave. The equation is implemented in the source code in (params.cpp, Parameters::Participate1) through (params.cpp, Parameters::Participate4).

Table 5
States

State Number	State Name	Class Name	Number to Add in Forming Path Variable
0	beginPaidLeaveElig	BeginPaidLeaveElig	1
1	beginPaidLeaveInelig	BeginPaidLeaveInelig	2
2	beginProgram	BeginProgram	4
3	beginUnpaidLeaveElig	BeginUnpaidLeaveElig	8
4	beginUnpaidLeaveInelig	BeginUnpaidLeaveInelig	16
5	continuePaidLeave	ContinuePaidLeave	32
6	continuePaidLeave2	ContinuePaidLeave_2	64
7	continueProgram	ContinueProgram	128
8	participate1	Participate_1	256
9	participate2	Participate_2	512
10	participate3	Participate_3	1,024
11	participate4	Participate_4	2,048
12	extend1	Xtend_1	4,096
13	extend2	Xtend_2	8,192
14	extend3	Xtend_3	16,384
15	extend4	Xtend_4	32,768
16	extend5	Xtend_5	65,536
17	receivePayAfterProgram	ReceivePayAfterProgram	131,072
18	unpaidLeaveToOL	UnpaidLeaveToOL	262,144
19	wait	Wait	524,288
20	endLeave	EndLeave	0

Notes: State Number refers to the numbers in Figures 3-5.
The source code is in the file benefitcalc.cpp and benefitcalc.h.

Figure 1
Paid Leave Simulation Flow

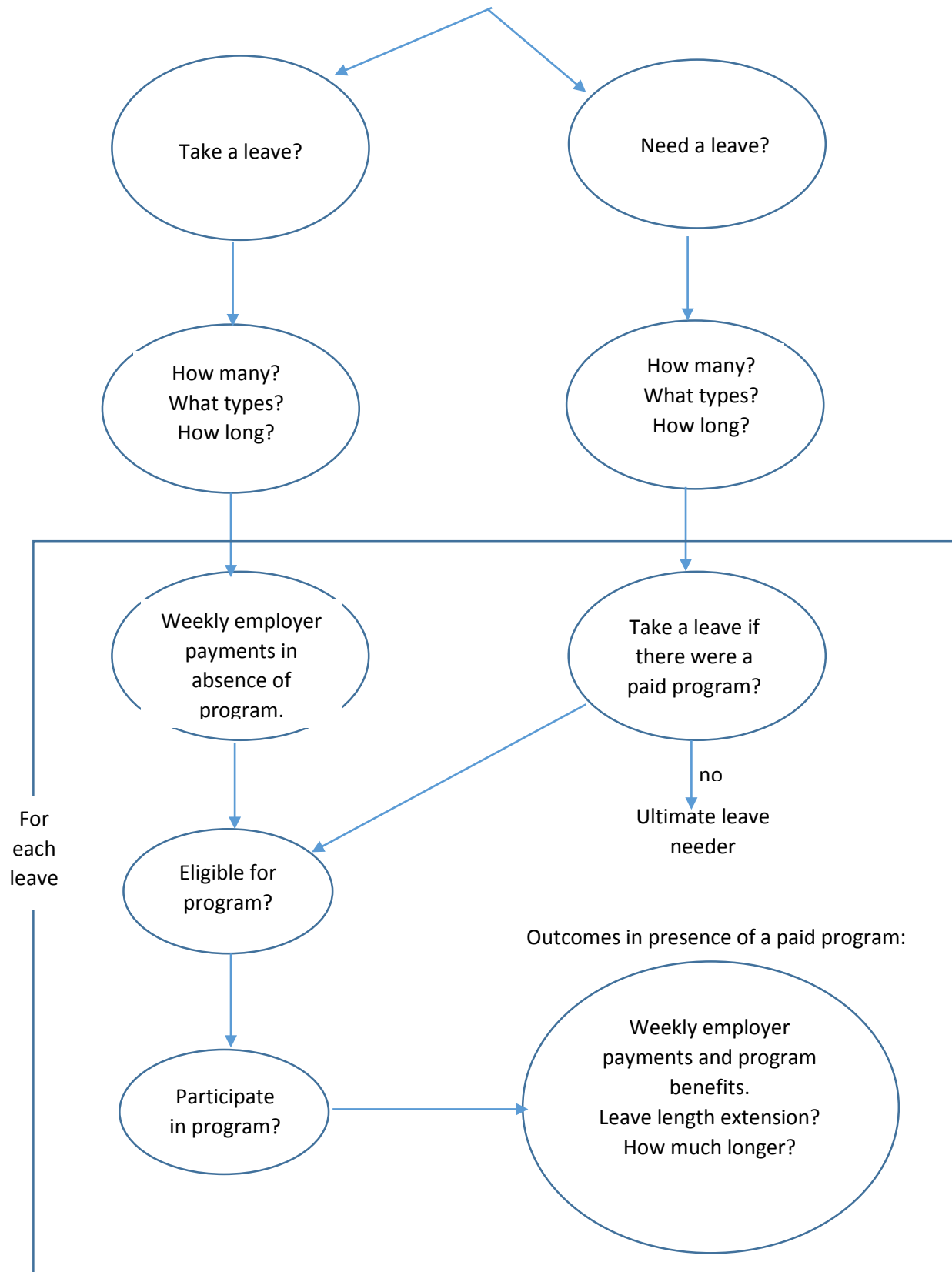


Figure 2
Weekly employer payments for leave in absence of a paid leave program

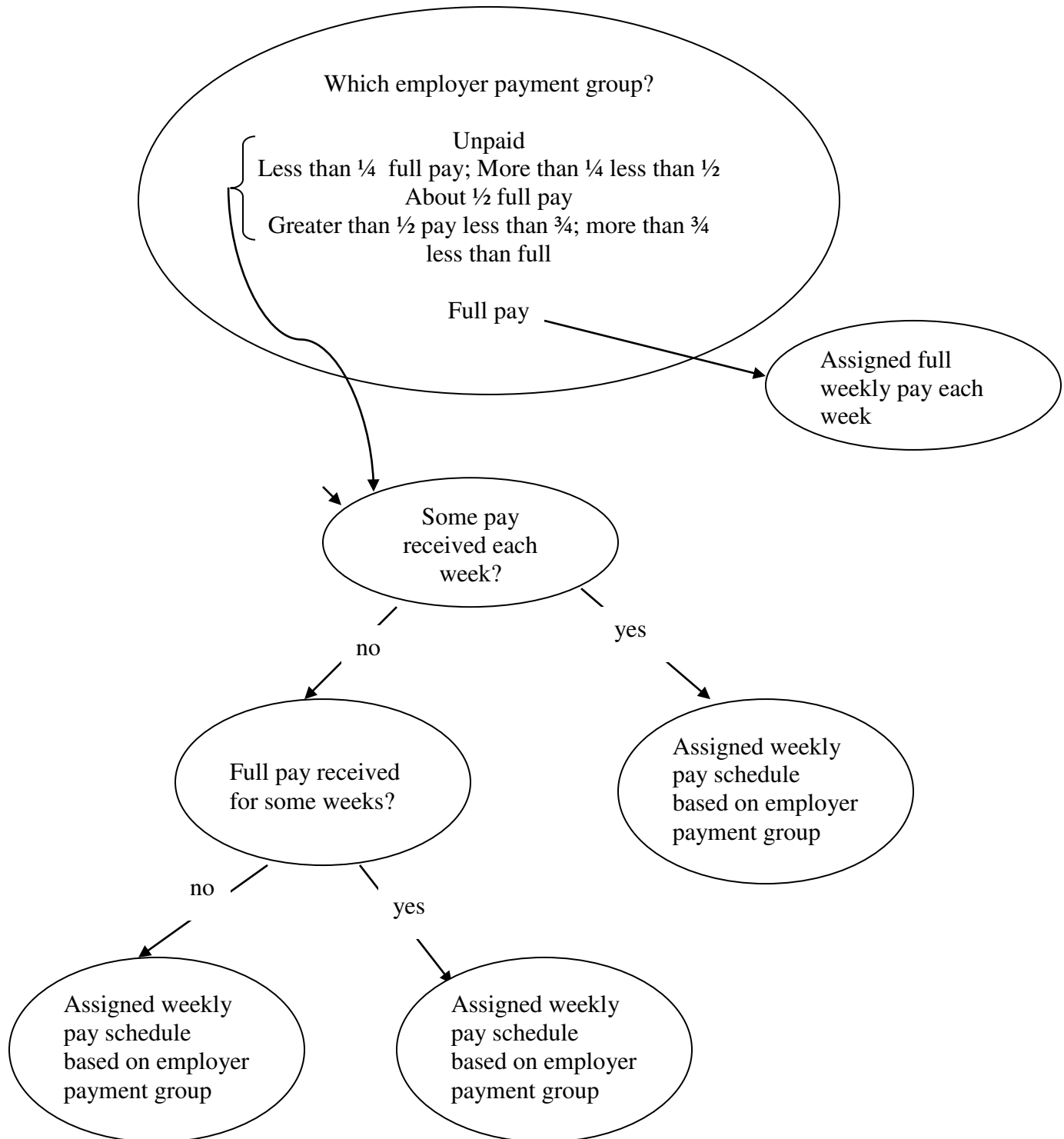
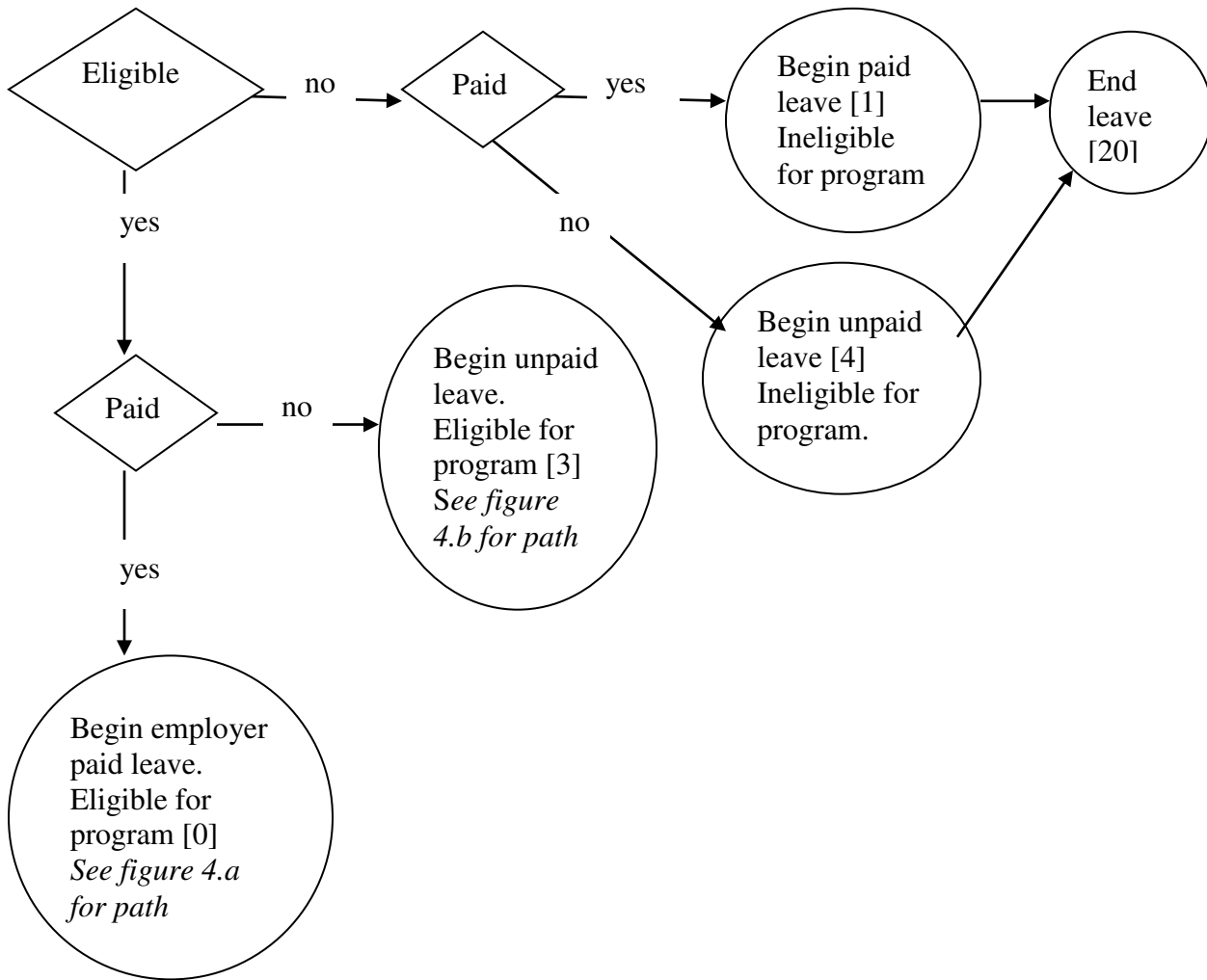


Figure 3
 Simulating Weekly employer pay and leave program benefits in presence of paid leave program



Note: Numbers in brackets in figures 3-5 correspond the state number in Table 5.

Figure 4.a
Simulating Use of Paid Leave Program and Employer Benefits for
Employees with Some Employer Paid Leave

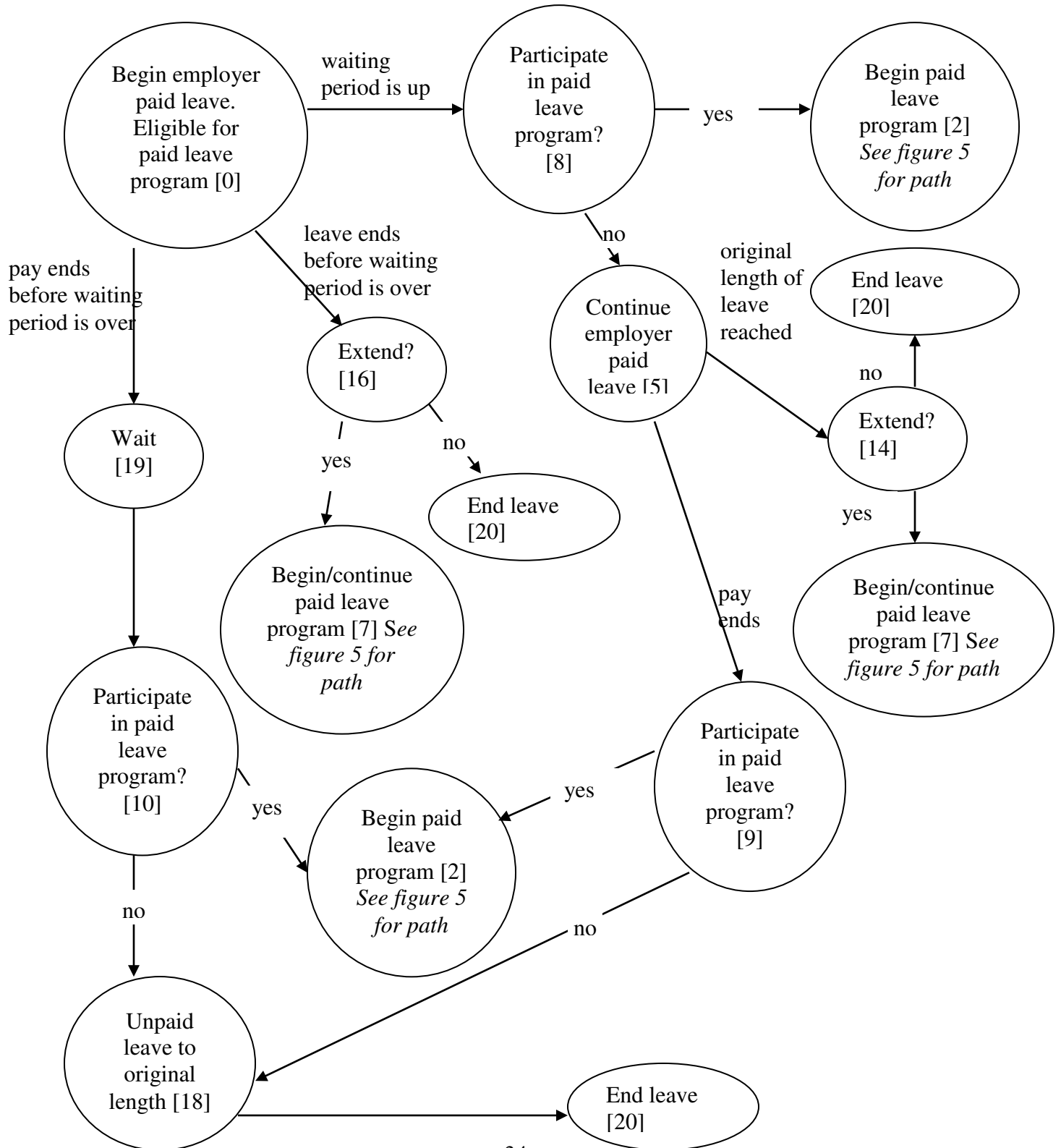


Figure 4.b
Simulating Use of Paid Leave Program for Employees with no employer
paid leave

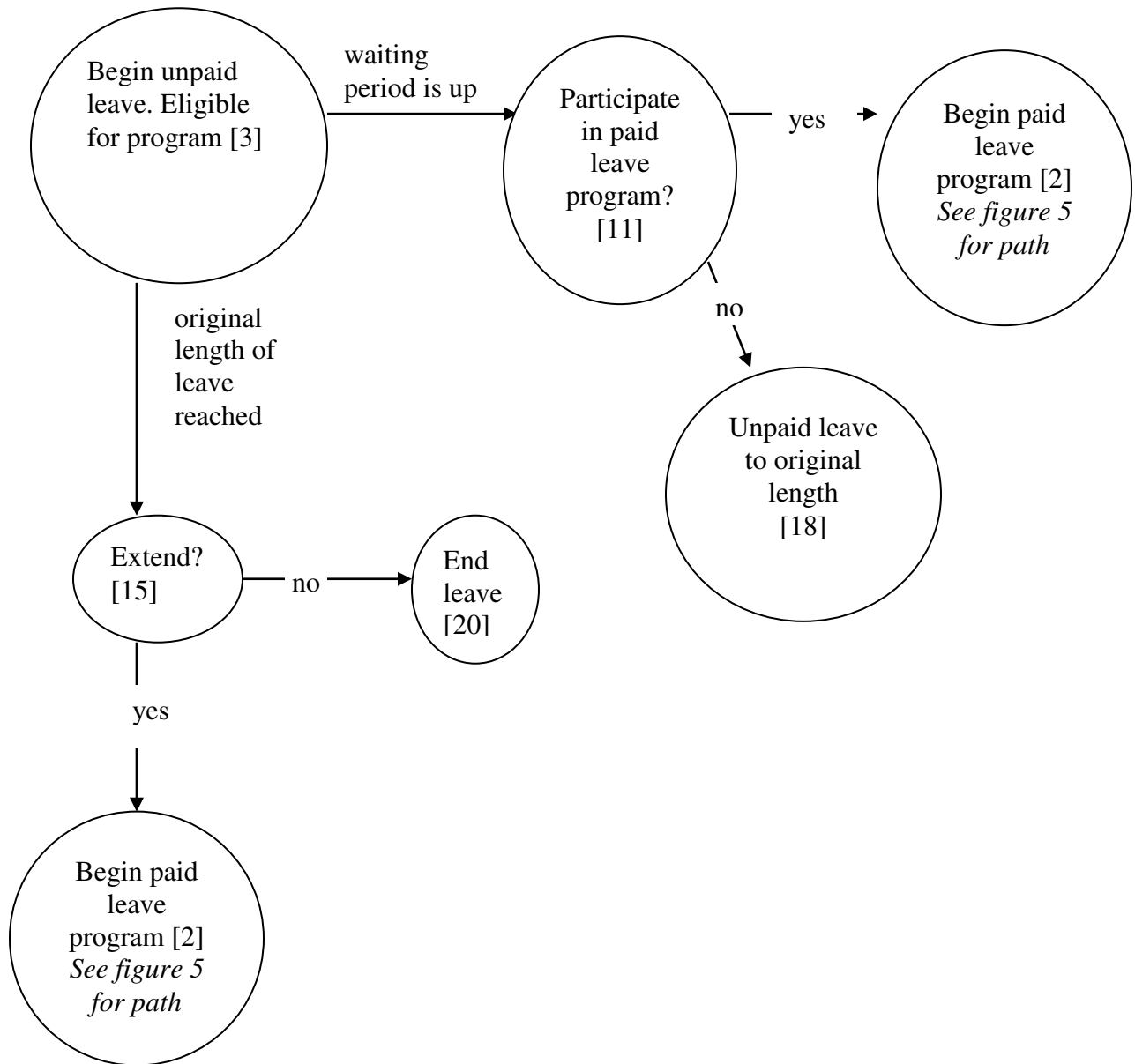
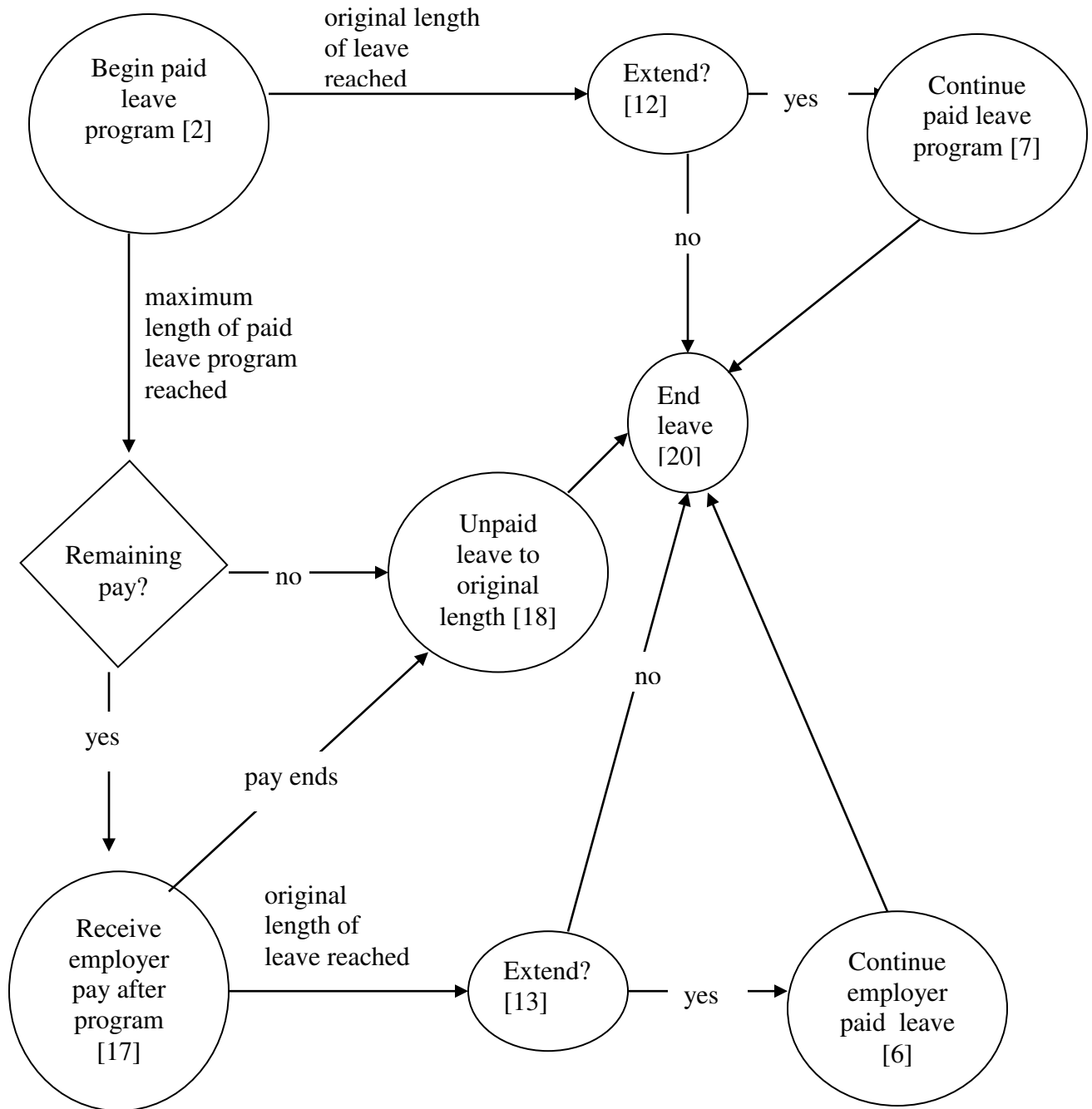


Figure 5
Simulating Use of Paid Leave Program Once it Begins



Appendix A

Sample Command File

```

*
* This is a sample command file
*
* The eligibility requirements are that workers must have
*   worked at least 1250 hours for a single employer in the last year.
* There is no employer size constraint.
* These eligibility rules are implemented by the eligibility rule MA_UIB.
*
* There is a waiting period of 1 week.
* Public employees are eligible.; but self-employed are not. (The default screen.)
* Benefits are tied to the average state weekly wage of $1,181.28:
*   95% if the wage is <=30% of the state average wage;
*   90% if the wage is >30% but <=50%;
*   80% if the wage is >50% but <=80%;
*   66% if the wage is >80% of the state average wage.
*   This is implemented by the FORMULA command instead of the REPLACEMENT RATIO command.
* There is a weekly benefit cap of $1,000.
* There is no dependent allowance.
* The program may extend a leave beyond what it would be in the absence of the program.
* The benefit amount affects participation.
* Maximum weeks depend on the leave type.  EDIT MY CHOICES
* Participation rates vary by leave type?  EDIT MY CHOICES
*
*
file log=log.txt debug=debug.txt main=main.csv
   leaves=leaves.csv weekly=weekly.csv states=states.csv benefit=weekb.csv emppay=weeke.csv
   doc=doc.txt;
file include=include.txt;
*file pumsh=ma_2009-2013_h.txt pumsp=ma_2009-2013_p.txt;
*file pumsh=ma_short_h.txt pumsp=ma_short_p.txt;
*file pumsh=us_short_h.txt pumsp=us_short_p.txt;
file pumsh=us_2009-2013_h.txt pumsp=us_2009-2013_p.txt;

benefiteffect Yes;
clonefactor 1;
*****dependentalallowance 25; No dependent allowance
*detail 7;
detail 2;
eligibilityrules type=MA_UIB;
extendleaves Yes;
fmlaprotectionconstraint no;
maxweeks MD=26 NC=26 default=12;
missingvalue ".";
*****replacementratio .5; Use FORMULA instead
formula yes;
takeuprates default=.667;
waitingperiod 1;
weeklybencap 1000;
weightfactor 1;
government yes;
*calibrate no;
calibrate yes;
*
* END
*

```

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
personid	person sequence number	0	software assigned	use this for identification and linking
leaveid	leave sequence number	0	software assigned	use this for identification and linking
weekid	week sequence number	0	software assigned	use this for identification and linking
weight	record weight	0	calculated from ACS	use this for weighting records to get population totals
MAIN				
insp		8	ACS	
rt		8	ACS	
serialno		1	ACS	
division		8	ACS	
puma00		2	ACS	
puma10		2	ACS	
region		8	ACS	
st		1	ACS	
adjhsg		8	ACS	
adjinc		2	ACS	
wgtp		3	ACS	
np		2	ACS	
type		8	ACS	
acr		8	ACS	
ags		8	ACS	
bath		8	ACS	
bdsp		8	ACS	
bld		8	ACS	
bus		8	ACS	
conp		8	ACS	
elep		8	ACS	
fs		8	ACS	
fulp		8	ACS	
gasp		8	ACS	
hfl		8	ACS	
mhp		8	ACS	
mrgi		8	ACS	
mrgp		8	ACS	
mrgt		8	ACS	
mrgx		8	ACS	
refr		8	ACS	
rmsp		8	ACS	
rntm		8	ACS	
rntp		8	ACS	
rwat		8	ACS	
rwatpr		8	ACS	
sink		8	ACS	
smp		8	ACS	
stov		8	ACS	
tel		8	ACS	
ten		8	ACS	
toil		8	ACS	
vacs		8	ACS	
valp		8	ACS	
veh		8	ACS	
watp		8	ACS	
ybl		8	ACS	
fes		2	ACS	
fincp		8	ACS	
fparc		2	ACS	
grntp		8	ACS	
grpip		8	ACS	
hhl		8	ACS	
hht		2	ACS	
hincp		8	ACS	
hugcl		8	ACS	
hupac		8	ACS	
hupaoc		8	ACS	
huparc		8	ACS	
kit		8	ACS	
lngi		8	ACS	
multg		8	ACS	
mv		8	ACS	
noc		2	ACS	
npf		2	ACS	
npp		8	ACS	
nr		8	ACS	
nrc		8	ACS	
ocpip		8	ACS	
partner		2	ACS	
plm		8	ACS	
psf		2	ACS	
r18		8	ACS	
r60		8	ACS	
r65		8	ACS	
resmode		8	ACS	
smocp		8	ACS	
smx		8	ACS	
srnt		8	ACS	
sval		8	ACS	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
taxp		8	ACS	
wif		8	ACS	
wkexrel		8	ACS	
workstat		8	ACS	
facrp		8	ACS	
fagsp		8	ACS	
fbathp		8	ACS	
fbdsp		8	ACS	
fbldp		8	ACS	
fbusp		8	ACS	
fconp		8	ACS	
felep		8	ACS	
ffsp		8	ACS	
ffulp		8	ACS	
fgasp		8	ACS	
fhflp		8	ACS	
finsp		8	ACS	
fkitp		8	ACS	
fmhlp		8	ACS	
fmrqip		8	ACS	
fmrqp		8	ACS	
fmrqtp		8	ACS	
fmrqxp		8	ACS	
fmvp		8	ACS	
fplmp		8	ACS	
frefrp		8	ACS	
frmsp		8	ACS	
frntmp		8	ACS	
frntp		8	ACS	
frwatp		8	ACS	
frwatprp		8	ACS	
fsinkp		8	ACS	
fsmp		8	ACS	
fsmxhp		8	ACS	
fsmxsp		8	ACS	
fstovp		8	ACS	
ftaxp		8	ACS	
ftelp		8	ACS	
ftenp		8	ACS	
ftoilp		8	ACS	
fvacsp		8	ACS	
fvalp		8	ACS	
fvehp		8	ACS	
fwatp		8	ACS	
fyblp		8	ACS	
wgtp1		8	ACS	
wgtp2		8	ACS	
wgtp3		8	ACS	
wgtp4		8	ACS	
wgtp5		8	ACS	
wgtp6		8	ACS	
wgtp7		8	ACS	
wgtp8		8	ACS	
wgtp9		8	ACS	
wgtp10		8	ACS	
wgtp11		8	ACS	
wgtp12		8	ACS	
wgtp13		8	ACS	
wgtp14		8	ACS	
wgtp15		8	ACS	
wgtp16		8	ACS	
wgtp17		8	ACS	
wgtp18		8	ACS	
wgtp19		8	ACS	
wgtp20		8	ACS	
wgtp21		8	ACS	
wgtp22		8	ACS	
wgtp23		8	ACS	
wgtp24		8	ACS	
wgtp25		8	ACS	
wgtp26		8	ACS	
wgtp27		8	ACS	
wgtp28		8	ACS	
wgtp29		8	ACS	
wgtp30		8	ACS	
wgtp31		8	ACS	
wgtp32		8	ACS	
wgtp33		8	ACS	
wgtp34		8	ACS	
wgtp35		8	ACS	
wgtp36		8	ACS	
wgtp37		8	ACS	
wgtp38		8	ACS	
wgtp39		8	ACS	
wgtp40		8	ACS	
wgtp41		8	ACS	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
wgtp42		8	ACS	
wgtp43		8	ACS	
wgtp44		8	ACS	
wgtp45		8	ACS	
wgtp46		8	ACS	
wgtp47		8	ACS	
wgtp48		8	ACS	
wgtp49		8	ACS	
wgtp50		8	ACS	
wgtp51		8	ACS	
wgtp52		8	ACS	
wgtp53		8	ACS	
wgtp54		8	ACS	
wgtp55		8	ACS	
wgtp56		8	ACS	
wgtp57		8	ACS	
wgtp58		8	ACS	
wgtp59		8	ACS	
wgtp60		8	ACS	
wgtp61		8	ACS	
wgtp62		8	ACS	
wgtp63		8	ACS	
wgtp64		8	ACS	
wgtp65		8	ACS	
wgtp66		8	ACS	
wgtp67		8	ACS	
wgtp68		8	ACS	
wgtp69		8	ACS	
wgtp70		8	ACS	
wgtp71		8	ACS	
wgtp72		8	ACS	
wgtp73		8	ACS	
wgtp74		8	ACS	
wgtp75		8	ACS	
wgtp76		8	ACS	
wgtp77		8	ACS	
wgtp78		8	ACS	
wgtp79		8	ACS	
wgtp80		8	ACS	
year		8	ACS	
extra1		8	ACS	
extra2		8	ACS	
extra3		8	ACS	
rt		8	ACS	
serialno		1	ACS	
sporder		1	ACS	
puma00		2	ACS	
puma10		2	ACS	
st		1	ACS	
adjinc		2	ACS	
pwgtp		2	ACS	
agep		2	ACS	
cit		5	ACS	
citwp05		8	ACS	
citwp12		8	ACS	
cow		2	ACS	
ddrs		8	ACS	
dear		8	ACS	
deye		8	ACS	
dout		8	ACS	
dphy		8	ACS	
drat		8	ACS	
dratx		8	ACS	
drem		8	ACS	
eng		5	ACS	
fer		2	ACS	
gcl		8	ACS	
gcm		8	ACS	
gcr		8	ACS	
hins1		8	ACS	
hins2		8	ACS	
hins3		8	ACS	
hins4		8	ACS	
hins5		8	ACS	
hins6		8	ACS	
hins7		8	ACS	
intp		8	ACS	
jwmnp		8	ACS	
jwrip		8	ACS	
jwtr		8	ACS	
lanx		8	ACS	
mar		2	ACS	
marhd		8	ACS	
marhm		8	ACS	
marht		8	ACS	
marhw		8	ACS	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
marhyp05		8	ACS	
marhyp12		8	ACS	
mig		8	ACS	
mil		8	ACS	
mlpa		8	ACS	
mlpb		8	ACS	
mlpcd		8	ACS	
mlpe		8	ACS	
mlpfg		8	ACS	
mlph		8	ACS	
mlpi		8	ACS	
mlpj		8	ACS	
mlpk		8	ACS	
nwab		5	ACS	
nwav		8	ACS	
nwla		8	ACS	
nwlk		8	ACS	
nwre		8	ACS	
oip		8	ACS	
pap		8	ACS	
relp		8	ACS	
retp		8	ACS	
sch		8	ACS	
schg		8	ACS	
schl		2	ACS	
semp		8	ACS	
sex		2	ACS	
ssip		8	ACS	
ssp		8	ACS	
wagp		8	ACS	
wkhp		8	ACS	
wkl		8	ACS	
wkw		8	ACS	
wrk		8	ACS	
yoep05		8	ACS	
yoep12		8	ACS	
anc		8	ACS	
anc1p05		8	ACS	
anc1p12		8	ACS	
anc2p05		8	ACS	
anc2p12		8	ACS	
decade		8	ACS	
dis		5	ACS	
drivesp		8	ACS	
esp		8	ACS	
esr		8	ACS	
fod1p		8	ACS	
fod2p		8	ACS	
hicov		8	ACS	
hisp		2	ACS	
indp		2	ACS	
jwap		8	ACS	
jwdp		8	ACS	
lanp05		8	ACS	
lanp12		8	ACS	
migpuma00		8	ACS	
migpuma10		8	ACS	
migsp05		8	ACS	
migsp12		8	ACS	
msp		8	ACS	
naicsp		2	ACS	
nativity		8	ACS	
nop		8	ACS	
oc		8	ACS	
occp02		2	ACS	
occp10		2	ACS	
occp12		2	ACS	
paoc		8	ACS	
pernp		2	ACS	
pincp		2	ACS	
pobp05		8	ACS	
pobp12		8	ACS	
povpip		2	ACS	
powpuma00		8	ACS	
powpuma10		8	ACS	
powsp05		8	ACS	
powsp12		8	ACS	
privcov		8	ACS	
pubcov		8	ACS	
qtrbir		8	ACS	
rac1p		2	ACS	
rac2p05		8	ACS	
rac2p12		8	ACS	
rac3p05		8	ACS	
rac3p12		8	ACS	
racaian		8	ACS	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
racasn		8	ACS	
racblk		8	ACS	
racnhpi		8	ACS	
racnum		8	ACS	
racsor		8	ACS	
racwht		8	ACS	
rc		8	ACS	
sciengp		8	ACS	
sciengrlp		8	ACS	
sfn		8	ACS	
sfr		8	ACS	
socp00		2	ACS	
socp10		2	ACS	
socp12		2	ACS	
vps		8	ACS	
waob		5	ACS	
fagep		8	ACS	
fancp		8	ACS	
fcitp		8	ACS	
fcitwp		8	ACS	
fcowp		8	ACS	
fddrsp		8	ACS	
fdearp		8	ACS	
fdeyep		8	ACS	
fdoutp		8	ACS	
fdphyp		8	ACS	
fdratp		8	ACS	
fdratxp		8	ACS	
fdremp		8	ACS	
fengp		8	ACS	
fesrp		8	ACS	
fferp		8	ACS	
ffodp		8	ACS	
fgclp		8	ACS	
fgcmp		8	ACS	
fgcrp		8	ACS	
fhins1p		8	ACS	
fhins2p		8	ACS	
fhins3c		8	ACS	
fhins3p		8	ACS	
fhins4c		8	ACS	
fhins4p		8	ACS	
fhins5c		8	ACS	
fhins5p		8	ACS	
fhins6p		8	ACS	
fhins7p		8	ACS	
fhisp		8	ACS	
findp		8	ACS	
fintp		8	ACS	
fjwdp		8	ACS	
fjwmnp		8	ACS	
fjwrip		8	ACS	
fjwtrp		8	ACS	
flanp		8	ACS	
flanxp		8	ACS	
fmarhdp		8	ACS	
fmarhmp		8	ACS	
fmarhtp		8	ACS	
fmarhwp		8	ACS	
fmarhyp		8	ACS	
fmarp		8	ACS	
fmigp		8	ACS	
fmigsp		8	ACS	
fmilpp		8	ACS	
fmilsp		8	ACS	
foccp		8	ACS	
foip		8	ACS	
fpap		8	ACS	
fpobp		8	ACS	
fpowsp		8	ACS	
fracp		8	ACS	
freip		8	ACS	
fretp		8	ACS	
fschgp		8	ACS	
fschlp		8	ACS	
fschp		8	ACS	
fsemp		8	ACS	
fsexp		8	ACS	
fssip		8	ACS	
fssp		8	ACS	
fwagp		8	ACS	
fwkhp		8	ACS	
fwklp		8	ACS	
fwkwp		8	ACS	
fwrkp		8	ACS	
fyoep		8	ACS	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
pwgtp1		8	ACS	
pwgtp2		8	ACS	
pwgtp3		8	ACS	
pwgtp4		8	ACS	
pwgtp5		8	ACS	
pwgtp6		8	ACS	
pwgtp7		8	ACS	
pwgtp8		8	ACS	
pwgtp9		8	ACS	
pwgtp10		8	ACS	
pwgtp11		8	ACS	
pwgtp12		8	ACS	
pwgtp13		8	ACS	
pwgtp14		8	ACS	
pwgtp15		8	ACS	
pwgtp16		8	ACS	
pwgtp17		8	ACS	
pwgtp18		8	ACS	
pwgtp19		8	ACS	
pwgtp20		8	ACS	
pwgtp21		8	ACS	
pwgtp22		8	ACS	
pwgtp23		8	ACS	
pwgtp24		8	ACS	
pwgtp25		8	ACS	
pwgtp26		8	ACS	
pwgtp27		8	ACS	
pwgtp28		8	ACS	
pwgtp29		8	ACS	
pwgtp30		8	ACS	
pwgtp31		8	ACS	
pwgtp32		8	ACS	
pwgtp33		8	ACS	
pwgtp34		8	ACS	
pwgtp35		8	ACS	
pwgtp36		8	ACS	
pwgtp37		8	ACS	
pwgtp38		8	ACS	
pwgtp39		8	ACS	
pwgtp40		8	ACS	
pwgtp41		8	ACS	
pwgtp42		8	ACS	
pwgtp43		8	ACS	
pwgtp44		8	ACS	
pwgtp45		8	ACS	
pwgtp46		8	ACS	
pwgtp47		8	ACS	
pwgtp48		8	ACS	
pwgtp49		8	ACS	
pwgtp50		8	ACS	
pwgtp51		8	ACS	
pwgtp52		8	ACS	
pwgtp53		8	ACS	
pwgtp54		8	ACS	
pwgtp55		8	ACS	
pwgtp56		8	ACS	
pwgtp57		8	ACS	
pwgtp58		8	ACS	
pwgtp59		8	ACS	
pwgtp60		8	ACS	
pwgtp61		8	ACS	
pwgtp62		8	ACS	
pwgtp63		8	ACS	
pwgtp64		8	ACS	
pwgtp65		8	ACS	
pwgtp66		8	ACS	
pwgtp67		8	ACS	
pwgtp68		8	ACS	
pwgtp69		8	ACS	
pwgtp70		8	ACS	
pwgtp71		8	ACS	
pwgtp72		8	ACS	
pwgtp73		8	ACS	
pwgtp74		8	ACS	
pwgtp75		8	ACS	
pwgtp76		8	ACS	
pwgtp77		8	ACS	
pwgtp78		8	ACS	
pwgtp79		8	ACS	
pwgtp80		8	ACS	
occp02i		8	ACS	
occp10i		8	ACS	
occp12i		8	ACS	
extrap1		8	ACS	
extrap2		8	ACS	
extrap3		8	ACS	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
wklywage	weekly wage	2	calculated from the ACS	Calculated from last years weeks worked and annual earnings, or a regression if earnings<=0.
wkwagely	weekly wage based on last years data	7	calculated from the ACS	Earnings last year divided by weeks worked.
wkwagere	weekly wage from a regression model if earnings <=0 on the ACS	7	simulated	weekly wage from a regression model if earnings<=0 on the ACS
wagesrc	source of wklywage	7	software assigned	2->annual data from last year; 3->regression model
hourly	1->paid hourly; 0->not	2	simulated	
prhourly	probability of being paid hourly	7	simulated	
employercat	employer size category	7	simulated	1-><10; 2->10-49; 3->50-99; 4->100-499; 5->500-999; 6->1000+
empsize	employer size	7	simulated	500->500+
numemployers	1->one employer last year; 2->employers last year; 3->3 or more	7	simulated	
familyincome	family income	2	simulated	
numdephchild18	number of dependent children under 18	2	calculated from the ACS	
weeksworked	weeks worked last 12 months	2	simulated	
majocc	Major Occupation	3	calculated from the ACS	
datesok	1->Dates of leaves do not overlap	7	simulated	
lvtype1	type of leave 1	1	simulated	1->own health, 2->maternity-disability, 3->new child, 4->ill child, 5->ill spouse, 6->ill parent
leavenum1	leave number 1	2	simulated	Counts leaves taken in the presence of a program.
neednum1	need leave number 1	2	simulated	Counts leaves needed in the absence of a program.
length1	length of leave in days 1	1	simulated	1 week = 5 days (only weekdays counted). length=lben+lpaid+lupaid
lben1	# of days receiving program benefits 1	1	simulated	1 week = 5 days (only weekdays counted)
lpaid1	# of days receiving employer pay 1	1	simulated	1 week = 5 days (only weekdays counted)
lupaid1	# of days unpaid (no benefits or pay) 1	1	simulated	1 week = 5 days (only weekdays counted)
dtbegn1	date leave began 1	3	simulated	days since January 1, 1960
dtbegd1	day leave began 1	7	simulated	
dtbegm1	month leave began 1	7	simulated	
dtbegy1	year leave began 1	7	simulated	
dtendn1	date leave ended 1	3	simulated	days since January 1, 1960
dtendd1	day leave ended 1	7	simulated	
dtendm1	month leave ended 1	7	simulated	
dtendy1	year leave ended 1	7	simulated	
trunc1	1->leave truncated, 0->not 1	1	simulated	leave ended after July 31, 2000
paid1	1->received some pay from employer, 0->not 1	1	simulated	
benefit1	1->received some program benefits, 0->not 1	1	simulated	
pdfull1	1->received full pay from employer, 0->not 1	1	simulated	
wkwage1	weekly wage 1	3	calculated from ACS	
lvweeks1	number of weeks on leave 1	7	calculated from days on leave	(days on leave)/5
payamt1	total payments received from employer while on this leave 1	3	simulated	
benamt1	total program benefits received while on this leave 1	3	simulated	
lostpd1	lost product 1	7	simulated	(weekly wage)x(number of weeks on leave)
uncoamt1	uncompensated amount 1	7	simulated	(lost product)-(payments from employer)-(program benefits)
paywk1	employer payments, first week on leave 1	7	simulated	The sum of paywkn and paywkin (includes changes in pay induced by program)
benwk1	benefits received, first week on leave 1	7	simulated	
doctor1	1->care of doctor required 1	3	simulated	1->yes, 0->no
hosptl1	1->in hospital overnight 1	3	simulated	1->yes, 0->no
elig1	1->eligible under program 1	3	simulated	1->yes, 0->no
need1	1->leave needed by not taken 1	2	simulated	
len_np1	length of leave in absence of program 1	2	simulated	The program may have induced a longer length of leave.
dtendnp1	date leave ended if no program 1	7	simulated	days since January 1, 1960
dtendnd1	day leave ended if no program 1	7	simulated	
dtendnm1	month leave ended if no program 1	7	simulated	
dtendny1	year leave ended if no program 1	7	simulated	
paywkn1	employer payments if no program, first week on leave 1	7	simulated	see paywk
paywkin1	employer payments induced by program, first week on leave 1	7	simulated	see paywk
lvtype2	type of leave 2	1	simulated	1->own health, 2->maternity-disability, 3->new child, 4->ill child, 5->ill spouse, 6->ill parent
leavenum2	leave number 2	2	simulated	Counts leaves taken in the presence of a program.
neednum2	need leave number 2	2	simulated	Counts leaves needed in the absence of a program.
length2	length of leave in days 2	1	simulated	1 week = 5 days (only weekdays counted). length=lben+lpaid+lupaid
lben2	# of days receiving program benefits 2	1	simulated	1 week = 5 days (only weekdays counted)
lpaid2	# of days receiving employer pay 2	1	simulated	1 week = 5 days (only weekdays counted)
lupaid2	# of days unpaid (no benefits or pay) 2	1	simulated	1 week = 5 days (only weekdays counted)
dtbegn2	date leave began 2	3	simulated	days since January 1, 1960
dtbegd2	day leave began 2	7	simulated	
dtbegm2	month leave began 2	7	simulated	
dtbegy2	year leave began 2	7	simulated	
dtendn2	date leave ended 2	3	simulated	days since January 1, 1960
dtendd2	day leave ended 2	7	simulated	
dtendm2	month leave ended 2	7	simulated	
dtendy2	year leave ended 2	7	simulated	
trunc2	1->leave truncated, 0->not 2	1	simulated	leave ended after July 31, 2000
paid2	1->received some pay from employer, 0->not 2	1	simulated	
benefit2	1->received some program benefits, 0->not 2	1	simulated	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
pdfull2	1->received full pay from employer, 0->not 2	1	simulated	
wkwage2	weekly wage 2	3	calculated from ACS	
lwweeks2	number of weeks on leave 2	7	calculated from days on leave	(days on leave)/5
payamt2	total payments received from employer while on this leave 2	3	simulated	
benamt2	total program benefits received while on this leave 2	3	simulated	
lostpd2	lost product 2	7	simulated	(weekly wage)x(number of weeks on leave)
uncoamt2	uncompensated amount 2	7	simulated	(lost product)-(payments from employer)-(program benefits)
paywk2	employer payments, first week on leave 2	7	simulated	The sum of paywknp and paywkin (includes changes in pay induced by program)
benwk2	benefits received, first week on leave 2	7	simulated	
doctor2	1->care of doctor required 2	3	simulated	1->yes, 0->no
hospitl2	1->in hospital overnight 2	3	simulated	1->yes, 0->no
elig2	1->eligible under program 2	3	simulated	1->yes, 0->no
need2	1->leave needed by not taken 2	2	simulated	
len_np2	length of leave in absence of program 2	2	simulated	The program may have induced a longer length of leave.
dtendnp2	date leave ended if no program 2	7	simulated	days since January 1, 1960
dtendnd2	day leave ended if no program 2	7	simulated	
dtendnm2	month leave ended if no program 2	7	simulated	
dtendny2	year leave ended if no program 2	7	simulated	
paywknp2	employer payments if no program, first week on leave 2	7	simulated	see paywk
paywkin2	employer payments induced by program, first week on leave 2	7	simulated	see paywk
lvtype3	type of leave 3	1	simulated	1->own health, 2->maternity-disability, 3->new child, 4->ill child, 5->ill spouse, 6->ill parent
leavenum3	leave number 3	2	simulated	Counts leaves taken in the presence of a program.
neednum3	need leave number 3	2	simulated	Counts leaves needed in the absence of a program.
length3	length of leave in days 3	1	simulated	1 week = 5 days (only weekdays counted). length=lben+lpaid+lupaid
lben3	# of days receiving program benefits 3	1	simulated	1 week = 5 days (only weekdays counted)
lpaid3	# of days receiving employer pay 3	1	simulated	1 week = 5 days (only weekdays counted)
lupaid3	# of days unpaid (no benefits or pay) 3	1	simulated	1 week = 5 days (only weekdays counted)
dtbegn3	date leave began 3	3	simulated	days since January 1, 1960
dtbegd3	day leave began 3	7	simulated	
dtbegm3	month leave began 3	7	simulated	
dtbegy3	year leave began 3	7	simulated	
dtendn3	date leave ended 3	3	simulated	days since January 1, 1960
dtendd3	day leave ended 3	7	simulated	
dtendm3	month leave ended 3	7	simulated	
dtendy3	year leave ended 3	7	simulated	
trunc3	1->leave truncated, 0->not 3	1	simulated	leave ended after July 31, 2000
paid3	1->received some pay from employer, 0->not 3	1	simulated	
benefit3	1->received some program benefits, 0->not 3	1	simulated	
pdfull3	1->received full pay from employer, 0->not 3	1	simulated	
wkwage3	weekly wage 3	3	calculated from ACS	
lwweeks3	number of weeks on leave 3	7	calculated from days on leave	(days on leave)/5
payamt3	total payments received from employer while on this leave 3	3	simulated	
benamt3	total program benefits received while on this leave 3	3	simulated	
lostpd3	lost product 3	7	simulated	(weekly wage)x(number of weeks on leave)
uncoamt3	uncompensated amount 3	7	simulated	(lost product)-(payments from employer)-(program benefits)
paywk3	employer payments, first week on leave 3	7	simulated	The sum of paywknp and paywkin (includes changes in pay induced by program)
benwk3	benefits received, first week on leave 3	7	simulated	
doctor3	1->care of doctor required 3	3	simulated	1->yes, 0->no
hospitl3	1->in hospital overnight 3	3	simulated	1->yes, 0->no
elig3	1->eligible under program 3	3	simulated	1->yes, 0->no
need3	1->leave needed by not taken 3	2	simulated	
len_np3	length of leave in absence of program 3	2	simulated	The program may have induced a longer length of leave.
dtendnp3	date leave ended if no program 3	7	simulated	days since January 1, 1960
dtendnd3	day leave ended if no program 3	7	simulated	
dtendnm3	month leave ended if no program 3	7	simulated	
dtendny3	year leave ended if no program 3	7	simulated	
paywknp3	employer payments if no program, first week on leave 3	7	simulated	see paywk
paywkin3	employer payments induced by program, first week on leave 3	7	simulated	see paywk
nleaves	number of leaves	1	simulated	
nneed_np	# leaves needed by not taken (without program)	3	simulated	These include leaves taken only because of program payments.
nneed	leaves needed but not taken	2	simulated	These are included in the variable nneed_np.
n_ownhea	number of leaves for own health	2	simulated	
n_matdis	number of leaves for maternity-disability	2	simulated	
n_newkid	number of leaves for a new child	2	simulated	
n_illkid	number of leaves for an ill child	2	simulated	
n_illspo	number of leaves for an ill spouse	2	simulated	
n_illpar	number of leaves for an ill parent	2	simulated	
covelig	1->meets employer size and work requirements for FMLA	3	simulated	
empelig	1->meets work requirements for FMLA	3	simulated	

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
eligwes	1->meets work and employer size requirements for the program; 0->not	3	simulated	
wklyben	weekly benefit amount	3	simulated	The weekly benefit amount under the program, disregarding whether or not the person is eligible. Note: wklyben=depben+wagerep.
depben	weekly dependent allowance	7	simulated	The weekly dependent allowance under the program, disregarding whether or not the person is eligible.
wagerep	weekly wage replacement	7	simulated	The weekly wage replacement under the program, disregarding whether or not the person is eligible.
dailyben	daily benefit allowance	7	simulated	The daily amount under the program, disregarding whether or not the person is eligible. Note: dailyben=wklyben/5.
LEAVES				
lvtype	type of leave	1	simulated	1->own health, 2->maternity-disability, 3->new child, 4->ill child, 5->ill spouse, 6->ill parent
leavenum	leave number	2	simulated	Counts leaves taken in the presence of a program.
neednum	need leave number	2	simulated	Counts leaves needed in the absence of a program.
length	length of leave in days	1	simulated	1 week = 5 days (only weekdays counted). length=lben+lpaid+lupaid
lben	# of days receiving program benefits	1	simulated	1 week = 5 days (only weekdays counted)
lpaid	# of days receiving employer pay	1	simulated	1 week = 5 days (only weekdays counted)
lupaid	# of days unpaid (no benefits or pay)	1	simulated	1 week = 5 days (only weekdays counted)
dtbegn	date leave began	3	simulated	days since January 1, 1960
dtbegd	day leave began	7	simulated	
dtbegm	month leave began	7	simulated	
dtbegy	year leave began	7	simulated	
dtendn	date leave ended	3	simulated	days since January 1, 1960
dtendd	day leave ended	7	simulated	
dtendm	month leave ended	7	simulated	
dtendy	year leave ended	7	simulated	
trunc	1->leave truncated, 0->not	1	simulated	leave ended after July 31, 2000
paid	1->received some pay from employer, 0->not	1	simulated	
benefit	1->received some program benefits, 0->not	1	simulated	
pdfull	1->received full pay from employer, 0->not	1	simulated	
wkwage	weekly wage	3	calculated from ACS	
lvweeks	number of weeks on leave	7	calculated from days on leave	(days on leave)/5
payamt	total payments received from employer while on this leave	3	simulated	
benamt	total program benefits received while on this leave	3	simulated	
lostpd	lost product	7	simulated	(weekly wage)x(number of weeks on leave)
uncoamt	uncompensated amount	7	simulated	(lost product)-(payments from employer)-(program benefits)
paywk	employer payments, first week on leave	7	simulated	The sum of paywknp and paywkin (includes changes in pay induced by program)
benwk	benefits received, first week on leave	7	simulated	
doctor	1->care of doctor required	3	simulated	1->yes, 0->no
hospitl	1->in hospital overnight	3	simulated	1->yes, 0->no
elig	1->eligible under program	3	simulated	1->yes, 0->no
need	1->leave needed by not taken	2	simulated	
len_np	length of leave in absence of program	2	simulated	The program may have induced a longer length of leave.
dtendnp	date leave ended if no program	7	simulated	days since January 1, 1960
dtendnd	day leave ended if no program	7	simulated	
dtendnm	month leave ended if no program	7	simulated	
dtendny	year leave ended if no program	7	simulated	
paywknp	employer payments if no program, first week on leave	7	simulated	see paywk
paywkin	employer payments induced by program, first week on leave	7	simulated	see paywk
dtpwbeg	Date program would begin	7	simulated	
dtfmia	Date FMLA protection would end	7	simulated	
wneeder	1->Was a needer; 0->not	7	simulated	
path	Path that the person took	4	simulated	
WEEKLY EMPLOYER PAY AND PROGRAM BENEFITS				
wkpay	weekly payment from employer	7	simulated	The sum of wkpayn and wkpayi (includes changes in pay induced by program).
wkben	weekly program benefits	7	simulated	
wkpayn	weekly payment from employer, if no program	7	simulated	See wkben
wkpayi	weekly payment from employer induced by program	7	simulated	See wkben
STATES				
state	Type of state	7	simulated	0->beginPaidLeaveElig; 1->beginPaidLeaveInelig; 2->beginProgram; 3->beginUnpaidLeaveElig; 4->beginUnpaidLeaveInelig; 5->continuePaidLeave; 6->continuePaidLeave2; 7->continueProgram; 8->participate1; 9->participate2; 10->participate3; 11->participate4; 12->extend1; 13->extend2; 14->extend3; 15->extend4; 16->extend5; 17->receivePayAfterProgram; 18->unpaidLeaveToOL; 19->wait; 20->endLeave
nstate	state sequence number	7	simulated	Used for ID/linking purposes
dstsbeg	Date state began	7	simulated	
dststend	Date state ended	7	simulated	
stdays	Number of days in state	7	simulated	Includes part days. Decision states take no time, except for extend4 and extend5.
WEEKLY BENEFITS				
nstate	state sequence number	7	simulated	Used for ID/linking purposes
wklyben	benefit this week	7	simulated	
benwkno	week number of benefit	7	simulated	
WEEKLY EMPLOYER PAYMENTS				
nstate	state sequence number	7	simulated	Used for ID/linking purposes

Appendix B

Contents of the Documentation Output File

VARIABLE NAME	LABEL	LEVEL	SOURCE	NOTES
wklywge	employer pay this week	7	simulated	
paywkno	week number of payment	7	simulated	
fracpay	1->a factional part of the pay received in the absense of the progam; 0->all this week's pre-program pay is recieved	7	simulated	Indicates whether or not the pay this week represents a fraction of the pay in the corresponding week while on leave in the pre-program counterfactual. Note: This does not indicate whether or not a fraction of the leave taker's full weekly pay was received.
daysp	# of days of this week in the corresponding state	7	simulated	

Command Syntax

The following notation is used to describe the syntax of the commands.

ABC	All items in uppercase are required. The spelling, but not the case, must match exactly.
<i>abc</i>	Italics represent a generic value that you replace with a specific value.
[abc]	The item <i>abc</i> is optional.
{abc}	The item <i>abc</i> may be repeated one or more times.
"abc"	The characters <i>abc</i> are required.
ab c	One of <i>a</i> , <i>b</i> , or <i>c</i> may be specified.
a::=b	The item <i>a</i> is defined in terms of <i>b</i> .

Notes:

- 1) Commands begin with a command name and end with a semicolon. The only exception is a comment, which is not really a command. Commands may use more than one line; each line of a multi-line comment must have an asterisk in the first column of the line.
- 2) The FILE, ELIGIBILITYRULES, MAXWEEKS, and TAKEUPRATES commands consist of one or more fields. These fields may appear in any order.
- 3) Commands are not case sensitive.
- 4) The commands may appear in any order; however, it is recommended that the FILE command that specifies the log file be the first command in the command file. Otherwise, you might have to inspect the *logtemp.txt* file for any error messages that were sent before the log file was opened. See the FILE command below.
- 5) The commands should be entered in a single text file and placed in the Input folder. When you launch the simulator, it will prompt you for the name of this file. It is recommended that you give the file the extension of ".txt"; and the extension is required when responding to the prompt from the simulator.

COMMENT

"*" *comment*

comment ::= any sequence of characters on a single line.

Example:

```
* This is a comment
```

Notes:

- 1) The "*" must appear in the first column of a line. The entire line is a comment.

- 2) Comments can appear anywhere in the command file.

FILE command

FILE { LOG | PUMSH | PUMSP | DEBUG | MAIN | LEAVES | WEEKLY | STATES | BENEFIT | EMPPAY | DOC | [INCLUDE] “=” *filespecification* } “;”

filespecification ::= a valid Windows file name, optionally in quotes, including a filename, and optional extension. If the specification includes spaces, it must be enclosed in quotation marks. Do not include the path to the filename (drive letters and/or folder paths).

Examples:

```
file log=log.txt pumsh=us_short_h.txt pumsp=us_short_p.txt
debug=debug.txt main=main.csv leaves=leaves.csv weekly=weekly.csv
states=states.csv benefit=weekb.csv emppay=weeke.csv doc=doc.txt
include=include.txt;
```

```
file log="my new log.txt";
```

Notes:

- 1) This command specifies the output and input files of the model. It is recommended that the first command in the command file specify the log file, which is where error messages will appear. Before a log file is specified, all messages to be logged go to a file named “*logtemp.txt*”, which will be created in the Output folder.
- 2) If the file name includes spaces, the file name must be enclosed in double quotes, as in the second example.
- 3) All files are required except for the INCLUDE file, which is optional.
- 4) The LOG, DEBUG, MAIN, LEAVES, WEEKLY, STATES, BENEFIT, EMPPAY, and DOC files are output files and will be created in the Output folder. If a file of the same name already exists in the Output folder, it will be overwritten by the newer file.
- 5) The PUMSH, PUMSP, and INCLUDE files are input files and must be placed in the Input folder. Since these are text files it is recommended that they have extensions of “.txt”.
- 6) The LOG, DEBUG, and DOC files are also text files and so it is recommended that they be given extensions of “.txt”.
- 7) The MAIN, LEAVES, WEEKLY, STATES, BENEFIT, and EMPPAY files are comma separated files, and so it is recommended that they be given extensions of “.csv”.
- 8) The LOG file contains messages useful for fixing errors in the command file, and also contains some messages about the status and successful or unsuccessful conclusion of the simulation run.

- 9) The INCLUDE file contains a list of variable names from the American Community Survey PUMS input files that you want included in the output MAIN file. Spell the names exactly as they appear in the ACS documentation, except that you must use lowercase.
- 10) The PUMSH and PUMSP files contain the names of the household and person PUMS comma-separated data files, as released by the Census Bureau. These should have “.csv” extensions. Since the content and variable order of these files changes from year to year, make sure you are using the appropriate version of the simulator. This version uses the 5-year U.S. PUMS files, 2009-2013. The contents of the PUMSH and PUMSP files for this Census ACS PUMS is given below. You may create extracts of these files and use them as input files if you wish, but if you want the extracts to sum to population totals, use the WEIGHTFACTOR command. Extracts must consist of entire rows of the PUMS files, which means that you can only extract sample households and the persons in them, not extracts consisting of a subsample of variables. The first example LOG command above uses a small extract of this 5-year ACS PUMS.

Contents of PUMSH=us_h.txt

```
ss13husa.csv
ss13husb.csv
ss13husc.csv
ss13husd.csv
```

Contents of PUMSP=us_p.txt

```
ss13pusa.csv
ss13pusb.csv
ss13pusc.csv
ss13pusd.csv
```

BENEFITEFFECT command

BENEFITEFFECT YES|NO “;”

Example:

```
benefiteffect Yes;
```

Notes:

- 1) Specifies whether the benefit amount affects participation in the program. If the command is not included, the default of “No” is applied, that is, that the benefit amount does not affect participation.

CLONEFACTOR command

CLONEFACTOR *positive_integer* “;”

positive_integer ::= a positive integer.

Example:

```
clonefactor 10;
```

Notes:

- 1) Specifies how many times each sample person will be “run” through the simulator. If the command is not specified, the default of “1” – that is, no cloning – is applied.

DEPENDENTALLOWANCE command

DEPENDENTALLOWANCE *real_number* “;”

real_number ::= a non-negative real number. Scientific notation (exponential notation) is not allowed.

Example:

```
dependentalallowance 25;
```

Notes:

- 1) This command sets the allowance for dependent children in dollars per week. If the command is not included, the default of “0” – no dependent allowance – is applied.

DETAIL command

DETAIL *positive_integer* “;”

Example:

```
detail 8;
```

Notes:

- 1) This command sets the level of detail – the number of variables – present in the output files. The level of detail can range from “1” – minimum detail, to “8” – full detail. If the command is not included, the default of “8” – full detail – is applied. This means, for example, that all variables from the ACS PUMS files will appear in the main output file. To restrict the list of variables from the PUMS files, choose a lower level of detail and use the INCLUDE file option in the FILE command to specify the ACS PUMS variables that you *do* want to appear in the main output file.

ELIGIBILITYRULES command

ELIGIBILITYRULES [TYPE “=” FMLA | MA_UIB] | [[A_EARNINGS “=” *real_number*] [B_WEEKS “=” *positive_integer*] [C_ANNHOURS “=” *positive_integer*] [D_EMPSIZE “=” *positive_integer*] [RULE “=” ALL | *logical_expression*]] “;”

logical_expression ::= A logical expression using the prefixes of the eligibility requirements used in the command (A,B,C,D) and the logical operators “&” and “|” for logical “and” and “or”.

Examples:

```
eligibilityrules type=fmla;  
eligibilityrules a_earnings=3000 c_annhours=1000 d_emptsize=50  
    rule=all;  
eligibilityrules a_earnings=3000 c_annhours=1000 d_emptsize=50;  
eligibilityrules a_earnings=3000 b_weeks=40 c_annhours=1000  
    d_emptsize=50 rule="A & (B|C) & D";  
eligibilityrules a_earnings=3000 b_weeks=40 c_annhours=1000  
    d_emptsize=50 rule=A&(B|C)&D;
```

Notes:

- 1) This command specifies the eligibility requirements for the paid leave program. You have the choice of using a built-in set of requirements – currently there are two: the eligibility requirements of the federal FMLA law or the Massachusetts UIB program – or a custom set of requirements based on earnings, weeks, hours, and/or employer size conditions.
- 2) If you choose a custom set of requirements, they represent: A) earnings (in dollars) in the last 12 months; B) weeks worked in the last 12 months; C) total number of hours worked in the last 12 months; and D) number of employees that work for the person’s employer, counting all locations where the employer operates.
- 3) In a customized set of requirements, the requirement can be that all specified conditions must be met, as in the second and third examples above; or that some other logical combination of conditions must be met, as in the fourth and fifth conditions above.
- 4) The second and third examples are identical, since if the RULE field is omitted, ALL is assumed.
- 5) If the logical expression contains spaces, it must be enclosed in double quotes, as in the fourth example above.
- 6) The fourth and fifth examples are identical. They require that, in order for a person to be eligible for the paid leave program, the person must have earned at least \$3,000 in the last 12 months; have either worked at least 40 weeks or at least 1,000 hours in the last 12 months; and must have worked for an employer who has at least 50 employees.
- 7) If this command is omitted, all workers are eligible for the paid leave program.

EXTENDLEAVES command

EXTENDLEAVES YES|NO “;”

Example:

```
extendleaves Yes;
```

Notes:

- 1) Specifies whether the presence of the paid leave program will, on average, lengthen leaves. If this command is not included, the default of “no” will be in effect, meaning that the average leave length (conditional on a person’s demographic characteristics) will not be affected by the program (although the propensity to take a leave might be affected).

MAXWEEKS command

MAXWEEKS [OH “=” *positive_integer*] [MD “=” *positive_integer*] [NC “=” *positive_integer*] [IC “=” *positive_integer*] [IS “=” *positive_integer*] [IP “=” *positive_integer*] [DEFAULT “=” *positive_integer*] “;”

Examples:

```
maxweeks OH=12 MD=6 NC=24 IC=7 IS=10 IP=20;  
maxweeks default=12;  
maxweeks MD=6 NC=24 IC=7 IS=10 default=12;
```

Notes:

- 1) This command sets the maximum number of paid weeks for each type of leave.
- 2) The types of leave are: OH (own health); MD (maternity disability); NC (new child); IC (ill child); IS (ill spouse); and IP (ill parent).
- 3) The default option sets the maximum number of paid weeks for the types of leave that are not specified in the MAXWEEKS command.
- 4) This command must be supplied. There are no default values.

MISSINGVALUE command

MISSINGVALUE *value* “;”

value ::= Any numeric or character value enclosed in double quotes.

Example:

```
missingvalue ".";
```

Notes:

- 1) This value will be used in output files when there is no value for that variable. For example, it would be used to indicate the leave length for a person who did not take a leave.

- 2) If this command is not supplied, a value of “.” (period) will be used. This is the default missing value in Stata for a numeric variable.

REPLACEMENTRATIO command

REPLACEMENTRATIO *real_number* “;”

Example:

```
replacementratio .5;
```

Notes:

- 1) This specifies the weekly benefit amount as a proportion of weekly pay.
- 2) If the command is not included, a default of 1.0 is applied, that is, the benefit is full pay.

TAKEUPRATES command

TAKEUPRATES [OH “=” *real_number*] [MD “=” *real_number*] [NC “=” *real_number*] [IC “=” *real_number*] [IS “=” *real_number*] [IP “=” *real_number*] [DEFAULT “=” *real_number*] “;”

Examples:

```
takeuprates OH=.8 MD=.9 NC=.85 IC=.7 IS=.6 IP=.6;
```

```
takeuprates default=.667;
```

```
takeuprates MD=.9 NC=.85 IC=.7 IS=.6 default=.667;
```

Notes:

- 1) This command sets the take-up rate for each type of leave, that is, the proportion of eligible leave takers who decide to use the program.
- 2) The take-up rates set by this command may be increased by the BENEFITEFFECT command.
- 3) The types of leave are: OH (own health); MD (maternity disability); NC (new child); IC (ill child); IS (ill spouse); and IP (ill parent).
- 4) The default option sets the take-up rate for the types of leave that are not specified in the TAKEUPRATES command.
- 5) This command must be supplied. There are no default values.

WAITINGPERIOD command

WAITINGPERIOD *integer* “;”

integer ::= an non-negative integer.

Example:

```
waitingperiod 1;
```


Notes:

- 1) The waiting period, in weeks, before paid benefits begin.
- 2) If this command is not included, the default value of “0” is applied, meaning that benefits begin on the first day of the leave.

WEEKLYBENCAP command

WEEKLYBENCAP *real_number* “;”

Example:

```
weeklybencap 500;
```

Notes:

- 1) This command sets the maximum weekly benefit paid by the program.
- 2) If this command is not included, the default of \$1,000,000 is applied, effectively meaning that there is no cap.

WEIGHTFACTOR command

WEIGHTFACTOR *real_number* “;”

Example:

```
weightfactor 1;
```

Notes:

- 1) This command multiplies the person weight on the ACS PUMS record.
- 2) Use this if you concatenate PUMS files together (the factor would be less than one) or if you subsample the PUMS file (the factor would be greater than one).
- 3) The final weight used in the simulator is the person weight on the ACS PUMS times this weighting factor, divided by the CLONEFACTOR value.
- 4) Do not use this if you are using a 2-, 3-, or 5-year PUMS product, since the Census Bureau has already applied a weighting factor to these files.
- 5) If you do not include this command, the default of “1” is applied.